



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**OVLÁDÁNÍ ROBOTICKÉHO RAMENE S VYUŽITÍM
ROZŠÍŘENÉ REALITY A TABLETU**

CONTROL OF ROBOT MANIPULATOR USING AUGMENTED REALITY AND TABLET

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MARTIN PRISTAŠ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÍTĚZSLAV BERAN, Ph.D.

BRNO 2018

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2017/2018

Zadání diplomové práce

Řešitel: **Pristaš Martin, Bc.**

Obor: Management a informační technologie

Téma: **Ovládání robotického ramene s využitím rozšířené reality a tabletu
Control of Robot Manipulator Using Augmented Reality and Tablet**

Kategorie: Uživatelská rozhraní

Pokyny:

1. Nastudujte problematiku rozšířené reality (AR) a realizace AR pro mobilní zařízení. Seznamte se s přístupy pro lokalizaci kamery v prostoru.
2. Navrhněte systém, který bude lokalizovat zobrazovací zařízení (mobilní zařízení) v prostoru, zobrazovat dostupné informace o scéně (pozice a typ objektů atd.) a umožní uživateli efektivně řídit/programovat robotické rameno.
3. Vyberte vhodné existující nástroje a navržený systém implementujte.
4. Řešení integrujte do robotického systému, který bude dodávat informace o scéně v reálném čase. Provedte experimenty na efektivitu a použitelnost řešení.
5. Vytvořte plakát a krátké video prezentující klíčové výsledky vašeho řešení.

Literatura:

- R. Hartson, P. Pyla. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*, Morgan Kaufmann, ISBN-10: 0123852412, 2012
- Dále dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a částečně body 3 a 4.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Beran Vítězslav, Ing., Ph.D.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2017

Datum odevzdání: 23. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
L.S. 602 66 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Cieľom diplomovej práce je vytvorenie experimentálnej aplikácie na manipuláciu virtuálnych objektov v rozšírenej realite pomocou tabletu pre ovládanie robotického ramena. Navrhnuté sú rôzne spôsoby manipulácie virtuálnych objektov pre ich transláciu, rotáciu a zmenu merítka. Tieto metódy sú otestované na niekoľkých užívateľoch a porovnané v rámci ich využiteľnosti. Aplikácia umožňuje odosielať pozíciu zmeny virtuálneho objektu robotickému ramenu PR2 a simulovať tak manipuláciu virtuálnych objektov vďaka rozšírenej realite.

Abstract

The aim of this thesis is to create an experimental application for manipulating virtual objects in the augmented reality using an tablet for controlling the robotic arm. There is created various ways of manipulating virtual objects for their translation, rotation, and scale change. These methods are tested on several users and compared within their usability. The application allows you to send the position of virtual object changes to the PR2 robot arm and simulate manipulation of virtual objects with augmented reality.

Klíčové slová

ARKit, Rozšířená realita, ROS, SceneKit, Visual Inertial Odometry, ARTable, iOS

Keywords

ARKit, Augmented reality, ROS, SceneKit, Visual Inertial Odometry, ARTable, iOS

Citácia

PRISTAŠ, Martin. *Ovládání robotického ramene s využitím rozšířené reality a tabletu*. Brno, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Vítězslav Beran, Ph.D.

Ovládání robotického ramene s využitím rozšířené reality a tabletu

Prehlásenie

Prehlasujem, že som diplomovú prácu vypracoval samostatne pod vedením Ing. Vítězslava Berana, Ph. D. V práci som uviedol všetky literárne zdroje z ktorých som čerpal informácie.

.....
Martin Pristaš
22. mája 2018

Podakovanie

Rád by som sa poďakoval vedúcemu práce pánovi Ing. Vítězslavovi Beranovi, Ph. D. za poskytnutie jeho odborných znalostí a myšlienok, ktoré mi pomohli pri realizovaní tejto práce. Taktiež by som sa rád poďakoval pánovi Ing. Michalovi Kapinusovi za pomoc a poskytnutie rád pri práci s ARTable a robotickým ramenom.

Obsah

1	Úvod	2
2	Rozšírená realita	3
2.1	Spôsob využitia	4
2.2	Visual Inertial Odometry	10
2.3	Spôsoby interakcie s 3D objektmi	13
2.4	ARTable - Pracovisko podporujúce rozšírenú realitu	15
2.5	Robotický operačný systém	16
2.6	ARKit - Knížnica pre rozšírenú realitu pre iOS	19
3	Návrh riešenia aplikácie pre ovládanie robotického ramena	23
3.1	Typické využitie a prínos	25
3.2	Požiadavky na systém	25
3.3	Módy systému	27
3.4	Metódy manipulácie s virtuálnymi objektmi	28
3.5	Navrhnuté grafické rozhranie	31
3.6	Komunikácia s robotickým ramenom PR2	33
4	Implementácia experimentálnej aplikácie	35
4.1	Technický popis a realizácia	35
4.2	Súradnicový systém a pracovná plocha	36
4.3	Implementácia módov systému	37
4.4	Implementácia metód pre manipuláciu virtuálnych objektov	40
5	Testovanie experimentálnej aplikácie	43
5.1	Popis a faktory testovania na užívateľoch	43
5.2	Testovanie	44
5.3	Zhodnotenie testovania	47
6	Záver	48
	Literatúra	49

Kapitola 1

Úvod

Cieľom diplomovej práce je vytvorenie experimentálnej aplikácie slúžiacej na manipuláciu virtuálnych objektov v rozšírenej realite pomocou tabletu, navrhnúť jej rozhranie a metódy ovládania objektov. Následne tieto metódy testovať s užívateľmi a získať tak cennú spätnú väzbu a iný pohľad na riešenie týchto problémov.

Rozšírená realita sa stáva čoraz častejšie populárnejšia vďaka prenosným zariadeniam, ktoré majú dostatočný výkon na to, aby v rozšírenej realite ponúkli užívateľovi graficky prepracované hry a zaujímavé aplikácie. Na začiatku som sa venoval rozboru rozšírenej reality, popísal jeden z princípov zaznamenávania priestoru, ako je chápaná v dnešnom svete a aké sú jej spôsoby využitia.

Do rozšírenej reality spadá aj vznikajúce pracovisko ARTable, ktoré je tvorené výskumnou skupinou Robo@FIT na vysokej škole Vysoké učení technické v Brně. Popíšem, z čoho sa pracovisko podporujúce rozšírenú realitu skladá a aké je jeho základné využitie. K popisu pracoviska je spojený aj popis robotického operačného systému (ROS), ktorý taktiež popíšem v tejto kapitole. Spolu s rozšírenou realitou popíšem aj novú knižnicu ARKit pre systém iOS podporujúcej rozšírenú realitu, ktorú využijem v implementácii tejto diplomovej práce. ARKit je knižnica, s ktorou sa pomerne jednoducho pracuje pri tvorbe rozšírenej reality, vytváraní a manipulácii s objektmi.

V návrhu riešenia experimentálnej aplikácie je uvedené jej typické využitie, prínos tejto aplikácie a časté problémy ktoré pri manipulácii s virtuálnymi objektmi nastávajú. Požiadavkami na systém som jasne definoval, ako má výsledná aplikácia fungovať. Výsledná aplikácia je zložená z niekoľkých módov a obsahuje rôzne metódy manipulácie s virtuálnymi objektmi. V závere kapitoly je navrhnuté užívateľské rozhranie experimentálneho systému a komunikácia s robotickým ramenom PR2.

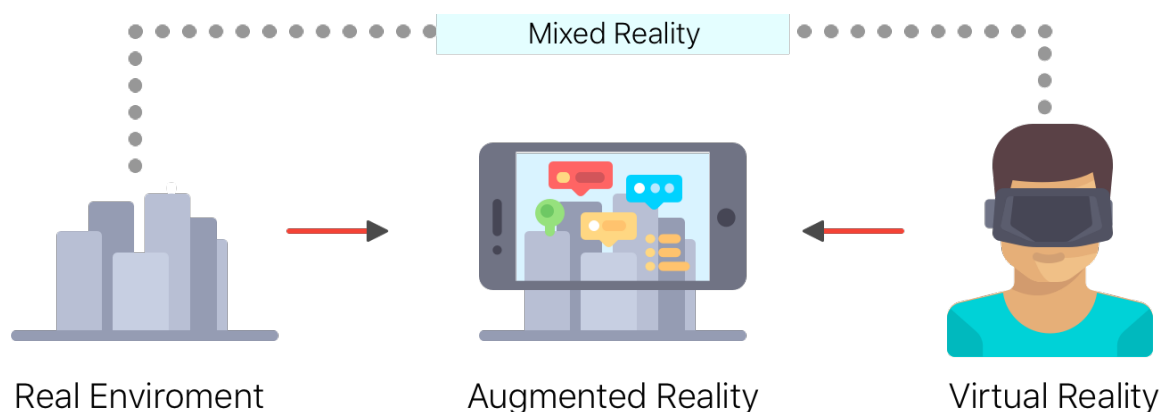
Implementácia aplikácie je realizovaná pre operačný systém iOS 11 na tablete Apple iPad 5th generation. V implementácii je popísané nastavovanie pracovnej plochy a súradnicového systému aplikácie a následne implementáciu módov systému so znázornením užívateľského rozhrania a aplikované metódy pre manipuláciu virtuálnych objektov z návrhu.

Testovanie aplikácie som rozdelil do dvoch častí, kde v prvej časti testujú aplikáciu užívatelia na študentskej konferencii Excel@FIT, ktorí nemajú informácie ako aplikácia má fungovať a ako sa jednotlivé módy aplikácie využívajú. Druhá časť testovania prebieha tak, že užívateľovi ukážem navrhnutú scénu, ktorú sa bude snažiť napodobniť a budem si všímať, ako s jednotlivými módmi aplikácie pracuje.

Kapitola 2

Rozšířená realita

Pod pojmom rozšířená realita si ešte stále množstvo ľudí predstavuje práve virtuálnu realitu. Rozšířená realita je pritom medzistupeň medzi realitou a virtuálnou realitou. Je to teda realita doplnená o virtuálne objekty práve pomocou napríklad mobilného telefónu v zaznamenanom reálnom prostredí pomocou kamery. Rozšířená realita tak má veľké množstvo využitia a posledné roky sa tak stáva čoraz populárnejšia aj práve vďaka jej dostupnosti na mobilných zariadeniach.



Obr. 2.1: Zaradenie pojmu rozšířená realita medzi realitu a virtuálnu realitu.[18]

Podľa vedeckého článku od R. T. Azuma (1997) [3] musí rozšířená realita spĺňať 3 základné charakteristiky:

- kombinovať virtuálny svet s reálnym,
- pracovať v reálnom čase,
- registrovať 3D prostredie.

Na základe tejto definície teda nie všetko môžeme považovať za rozšírenú realitu. Často si ľudia predstavia filmy s vizuálnymi efektami ako napríklad *Jurský Park* alebo *Avatar*, ktoré predstavujú doplnené reálne prostredie virtuálnymi objektami. To však nespĺňa definíciu rozšírenej reality, nakoľko film je statický a nie je nijak interaktívny.

V definícii nie je presne špecifikované výstupné zariadenie ako napríklad displej umiestnený na hlave, alebo nijak neobmedzuje audiovizuálne médiá. Definícia vyžaduje riadenie

v reálnom čase a priestorový záznam, čo znamená presné zarovnanie zodpovedajúcich virtuálnych a reálnych informácií v reálnom čase. Používateľ môže virtuálne objekty meniť a inak upravovať v prostredí, ktoré si stále zachováva referenčnú pozíciu voči reálnemu prostrediu. Používateľ prechádza po reálnom prostredí, systém získava informácie o prostredí a kombinuje tak virtuálny svet s reálnym.

2.1 Spôsob využitia

Vďaka tomu, že sa rozšírená realita vyvíja posledné roky dosť intenzívne, získala množstvo spôsobov využitia a častokrát si ani neuvedomujeme, že niektoré prvky z bežného života s ktorými sa stretávame patria priamo do rozšírenej reality. História rozšírenej reality je celkom bohatá s prvými pokusmi priblížiť ju človeku a efektívne využiť to, čo ponúka. Reálne využitie našla skôr v posledných rokoch príchodom prenosných zariadení a výkonnejšieho hardware.

Priemysel

Rok 1992 je označovaný ako rok vzniku rozšírenej reality, keď sa prvý krát v spoločnosti Boeing zobrazila schéma zapojenia kábeláže pomocou displeja pripevneného na hlave užívateľa pred očami. Toto otvorilo dvere rozšírenej reality do priemyslu a hľadali sa rôzne spôsoby využitia.



Obr. 2.2: Využitie rozšírenej reality v priemysle. Prevzaté z [12].

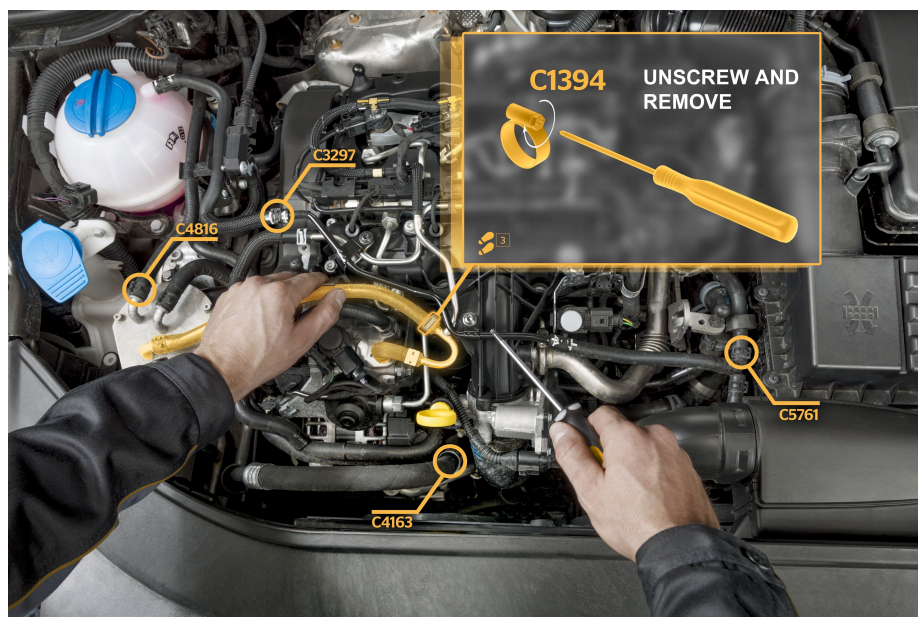
Architektonické štruktúry, infraštruktúra alebo aj stroje sú navrhované pomocou CAD programov, avšak veľa úprav a nepredvídaných vecí nastane až pri reálnom nasadení navrhovaných vecí. V tomto smere sme tiež našli uplatnenie rozšírenej reality, keď pomocou špeciálnych prístrojov na mapovanie 3D prostredia vieme identifikovať prostredie a virtuálne do neho dosadiť napríklad obrovské stroje alebo komplexné architektonické riešenia. To, že virtuálne dosadzujeme rôzne objekty, môžeme im nastaviť priehľadnosť alebo ich inak modifikovať v reálnom čase nám umožňuje pohodlne získavať tak presnejšiu predstavu reálneho nasadenia.

Veľmi dobré a efektívne využitie rozšírenej reality je aj v hľadaní potrubí pod zemou na základe GPS dát a známych záznamov kde sa potrubia nachádzajú. Uľahčí to tak výkopové práce alebo hľadanie poruchových častí pod zemou bez zbytočnej práce navyiac.

Samozrejme, v priemyselnej časti má rozšírená realita obrovské využitie a to najmä v továrňach, ktoré využívajú množstvo robotických ramien a počítačovo riadených strojov. Tu sa nájde uplatnenie či už pri oprave zložitých strojov, ich nastavovaní, prípadne rôznych upozorneniach pre obsluhujúceho pracovníka alebo rôznych návodoch ako daný stroj obsluhovať.

Údržba a technické porozumenie

Pre mnoho profesionálov je výzvou pochopiť ako niektoré veci fungujú, ako ich zložiť, rozobrať alebo opraviť. Inžinieri väčšinou strávia veľkú časť svojho času študovaním manuálov, technických nákresov a dokumentácií ktoré nie je možné si zapamätať do detailov v takom množstve. Rozšírená realita aj v tomto nájde svoje uplatnenie, či už je to pomoc pri výmene oleja v aute a navigovanie užívateľa pomocou nej, alebo oprava malého spotrebiča v domácnosti. Nakoľko sa v súčasnosti málo stretávame s takouto aplikáciou rozšírenej reality, myslím si, že čoskoro sa to postupne rozšíri aj v tomto spôsobe použitia.



Obr. 2.3: Spôsob využitia rozšírenej reality pri opravovaní závady v aute. Prevzaté z [2].

Medicína

Zaujímavé využitie rozšírenej reality nájdeme aj v medicíne, nakoľko nám umožňuje simulovať a prezentovať rôzne informácie potrebné v reálnom čase napríklad pri operácii. S využitím v medicíne sa môžeme stretnúť už pri aplikáciách dostupných na mobilné zariadenia, kde si užívateľ aplikácie môže zobrazíť orgány ľudského tela, simulovať tieto časti ako majú správne pracovať prípadne ako pracujú v prípade že nie sú zdravé. Štúdia, ktorá sa venuje rozšírenej realite v vzdelávacej oblasti ukázala, že metóda učenia je veľmi efektívna a

nahradzuje tak množstvo praxe, nižšiu mieru zlyhania, vyššiu presnosť výkonu, zrýchlené učenie a tiež lepšie pochopenie spojitostí v ľudskom tele.

Rozšírená realita má potenciál poskytnúť lekárom reálnejšie tréningové skúsenosti, čo vedie k lepším operáciám. Lekári, ktorí objavujú nové postupy, môžu tiež využiť ponuky zobrazovania anatómie v rozšírenej realite. Aj napriek tomu, že rozšírená realita má veľa vzdelávacích prínosov, štúdia tiež poukázala na niekoľko nevýhod, vrátane nedostatku učebných teórií dostupných na usmernenie designu rozšírenej reality a že tradičné učebné teórie sa nezhodujú so štýlom učenia pomocou rozšírenej reality.

V medicínskom prostredí sa jedná o široké spektrum využitia rozšírenej reality, či už je to na vzdelávacie účely pre budúcich doktorov, alebo na rôzne vizualizácie ľudského tela, prípravy na operáciu, ale aj napríklad plastická chirurgia kde sa môže zobrazovať výsledný efekt plastickej operácie.



Obr. 2.4: Aplikovanie rozšírenej reality v medicíne. Prevzaté z [21].

Zobrazenie informácií

Vďaka tomu, že sa rozšírená realita dostala do mobilných zariadení ktoré sú v dnešnej dobe pomerne výkonné a dokáže využiť spojenie toho, čo mobilné zariadenie ponúka – GPS dáta, dáta z akcelerometra, kompas a najmä kameru je jej využitie vo veľkom množstve aplikácií úplne bežné a atraktívne.

Či už sú to aplikácie ako *Snapchat* alebo *Messenger*, ktoré napríklad dopĺňajú užívateľovú tvár o rôzne efekty v reálnom čase, alebo rôzne hry, utility, navigácie. Rozšírená realita na mobiloch je čoraz viac populárnejšia a v mnohých prípadoch užitočná.

Jednou z prvých aplikácií ktoré dokazovali to, ako veľmi je rozšírená realita na mobiloch užitočná bola aplikácia *Translator* od spoločnosti *Google Inc.*, ktorá v reálnom čase prekladala text zachytený kamerou do požadovaného jazyka viď obrázok 2.5.



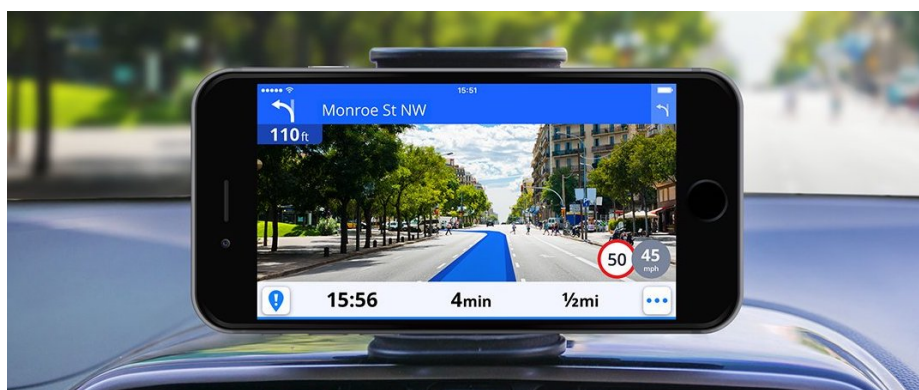
Obr. 2.5: Prekladanie textu zachytenom kamerou v reálnom čase. Prevzaté z [22].

Navigácia

Prvé využívanie rozšírenej reality v navigácii bolo používanie heads-up displeja vo vojenskom lietadle. Tieto displeje spočiatku však ukazovali len základné informácie, ako napríklad rýchlosť alebo krútiaci moment. Neskôr táto technológia prišla aj do áut, kde zobrazovala na čelnom skle informácie pre vodiča o vozidle, rýchlosti a navigáciu.

Asi najznámejším použitím v navigácii si môžeme predstaviť palubný počítač auta, ktorý zobrazuje to, kam bude vodič cúvať. Vykresľuje teda trajektóriu cúvania do obrazu z cúvacej kamery v reálnom čase. Toto použitie je veľmi známe, avšak málo kto si uvedomuje že sa jedná práve o použitie rozšírenej reality.

Ďalším možným použitím je spojenie GPS dát a kamery v telefóne, kde priniesla napríklad navigácia *Sygic* možnosť navigovania pomocou rozšírenej reality tak, že na obraze telefónu je vidieť vyznačenú cestu z navigovania.



Obr. 2.6: Navigácia Sygic a využitie rozšírenej reality. Prevzaté z [14].

Televízia

Svoje uplatnenie si našla aj v televízi, kde tiež častokrát nevieme, že sa jedná práve o rozšírenú realitu. Väčšinou je aplikovaná na športové podujatia ako vyznačovanie hráčov na ihrisku alebo živé vysvetlivky počas prenosu, vypisovania skóre a podobne. Taktiež si môžeme predstaviť predpoveď počasia, kde je osoba postavená pred zelené plátno ktoré sa filtruje v postprodukcii a do obrazu sa dopĺňa virtuálny obsah.



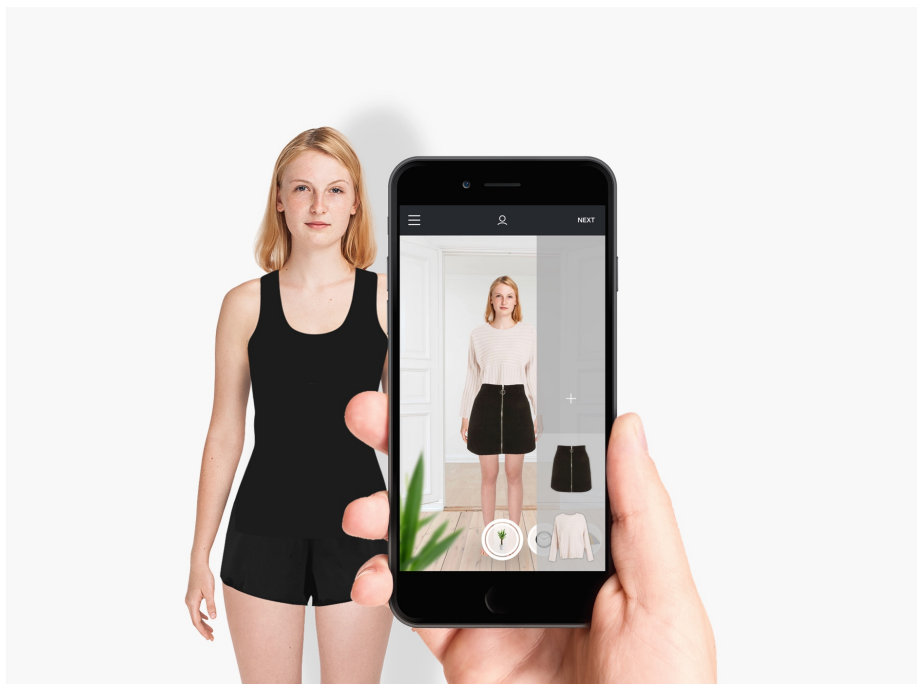
Obr. 2.7: Rozšírená realita použitá pri vysielaní zápasu hádzanej. Prevzaté z [19].

Reklama a marketing

Možnosť zobrazenia produktov potencionálnemu zákazníkov prostredníctvom rozšírenej reality je čoraz viac atraktívnejšia. Táto technológia môže viesť k skutočne interaktívnym zážitkom pre zákazníkov. Napríklad v obchode s Legom zákazník môže zobrať krabicu s hračkou, načítať si tam kód z krabice a zobraziť pomocou AR postavený model v reálnej veľkosti.

Niekedy je možné použiť rozšírenú realitu aj pre magazíny alebo letáky, kde užívateľ použije kameru telefónu pri prehliadaní časopisu a na displeji telefónu sa mu môže zobraziť rôzny nový obsah, napríklad statické obrázky môžu byť nahradené rôznymi animáciami a podobne.

Šikovná aplikácia *Pictofit* umožňuje virtuálne obliecť užívateľovi oblečenie ktoré ponúka aby zistil, či mu dané oblečenie sedí alebo nie viď obrázok 2.8.



Obr. 2.8: Aplikácia *Pictofit* ponúkajúca oblečenie ktoré si užívateľ môže vyskúšať v rozšírenej realite. Prevzaté z [9].

Hry

Jedným z prvých komerčných využití rozšírenej reality v hrách bola hra *The Eye of Judgment*, interaktívna hra pre Sony PlayStation 3. Hranie hry vyžadovalo použitie kamery umiestnenej na hlave. Hranie hier v rozšírenej realite v porovnaní s hraním video hier predstavuje úplne odlišné spôsoby hrania, kde deti môžu premeniť celú miestnosť na hráčske pole, hýbať sa, skákať alebo schovávať. Niektoré hry dokážu zmapovať v 3D reálne prostredie a premeniť ho na hráčske pole. Množstvo nových hier vzniká pre mobilné zariadenia, kde interaktivita s hraním hry v rozšírenej realite je oveľa vyššia a záživnejšia ako hranie hier iba na displeji tabletu alebo telefónu.



Obr. 2.9: Interaktívna hra Lego s využitím rozšírenej reality. Prevzaté z [4].

2.2 Visual Inertial Odometry

Presná technológia akú firma Apple používa vo svojej knižnici ARKit bohužiaľ nie je známa, prezradila len toľko, že sa jedná o Visual Inertial Odometry. VIO používa ako zdroj dát kameru zariadenia spolu s využitím senzoru na zaznamenávanie pohybu (anglicky inertial measurement unit, IMU). Sensory IMU integrujú multiosové kombinácie presných gyroskopov, akcelerometrov, magnetometrov a snímačov tlaku, aby poskytli spoľahlivé stanovenie polohy a pohybu pre stabilizačné a navigačné aplikácie.

Veľmi častou a populárnou metódou pre vizuálne zaznamenávanie priestoru je metóda SLAM (Simultaneous Localization and Mapping). Táto metóda sa využíva najmä v mapovaní priestoru v robotike alebo v autonómnych autách. Môže byť rozšírená o využitie vizuálnych dát z duálnej kamery, neurónové siete, IMU alebo hĺbkových ultrazvukových a infračervených senzorov.

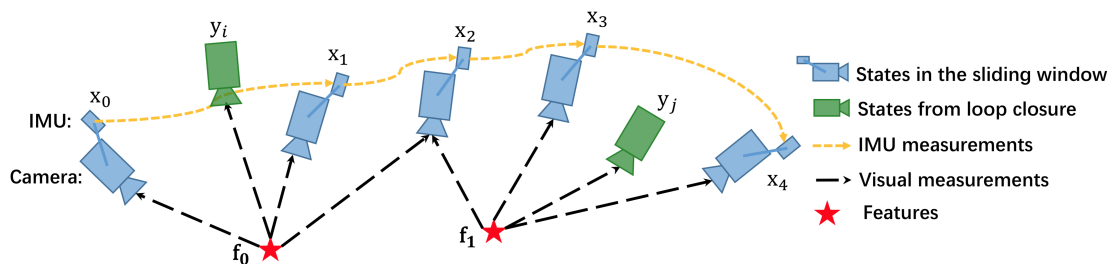
Keďže niektoré podporované zariadenia iPhonov a iPadov nemajú duálnu kameru, ARKit pracuje len s využitím jednej kamery. Taktiež v telefóne chýba akýkoľvek hĺbkový senzor pomocou ktorého by bolo jednoduchšie mapovanie priestoru a tak sa musíme spoliehať len na vypočítané dáta z vizuálnych a dopočítaných dát spojených s IMU. Dá sa však počítať s tým, že Apple sa bude snažiť v budúcnosti použiť duálnu kameru iPhone, prípadne doplniť iPhone o hĺbkový senzor.

SLAM, alebo inak povedané aj Visual odometry, využíva pokročilé 6-osé (Six degrees of freedom – **6DoF**) sledovanie pozície kamery na základe ľubovoľného počiatočného bodu. Visual odometry vypočítava rekonštrukciu 3D prostredia s využitím prírastkového sledovania (incremental tracking).

Pozn. 6DoF – v 3 dimenzionálnom priestore využíva 3 osi pre pohyb a 3 osi pre rotáciu.

Základný algoritmus SLAM [18] využíva postupne nasledujúce kroky:

1. Detekcia zaujímavých bodov z prvého obrazu – môže využívať napríklad algoritmy **Harris** alebo **FAST corners** popísané v 2.2.
2. Sledovanie zaujímavých bodov v 2D z predchádzajúceho obrazu – môže využívať algoritmus **KLT** popísaný v 2.2.
3. Určenie posuvnej matice medzi aktuálnym a predchádzajúcim obrazom pomocou algoritmu piatich bodov – anglicky **Five-Point Algorithm for Essential Matrix** popísaný v 2.2.
4. Upraviť pozíciu kamery v 3D priestore pomocou posuvnej matice.
5. Pre výpočet celej transformačnej matice treba využiť dáta z IMU, nakoľko pri práci v 2D obrazmi je známy pohyb kamery len v 2D priestore. Táto technika dopočítavania sa volá anglicky **Structure from motion (SFM)**¹.
6. Spracovanie nasledujúceho obrazu.



Obr. 2.10: Jednoduchá predstava o algoritme SLAM, detekovaní významných bodov, využitie IMU a detekovanie slučiek v obraze.[10]

Algoritmus SLAM s veľkou pravdepodobnosťou využíva aj knižnica ARKit, ktorá pracuje tak, že po zapnutí kamery začína detekovať významné body z obrazu a počas tejto detekcie je potrebné hýbať so zariadením v rôznych smeroch kvôli kalibrácii. Pri debugovaní² ARKit-u si môžeme zapnúť zobrazenie významných bodov **angl. feature points** pomocou nastavenia premennej `showFeaturePoints`. ARKit umožňuje pri debugovaní zapnúť pomocou premennej `showWorldOrigin` aj zobrazenie súradnicovej osi po detekovaní priestoru, ktorá zobrazuje polohu a orientáciu súradnicového systému rozšírenej reality.

Algoritmus Harris Corners a FAST

Harris Corners

Vzhľadom na to, že obraz má 2D rozmer, detekovanie záujmových bodov musí mať silný gradient vo vertikálnom, aj horizontálnom smere. Takže vhodné záujmové body budú vo všeobecnosti tvarované ako kruhové bloby alebo rohy. Detekovanie algoritmom Harris [6] využíva na stanovenie rohov obrazovú autokorelačnú maticu.

Formulácia detekcie rohov je založená na autokorelačnej matici $A(x, y)$, ktorá popisuje gradient W v okolí bodu $I(x, y)$ s gradientom bodu v obraze posunutým:

¹https://en.wikipedia.org/wiki/Structure_from_motion

²<https://developer.apple.com/documentation/arkit/arscndebugoptions>

$$A(x, y) = \begin{bmatrix} \sum_{i,j} I_x(x+i, y+j)^2 & \sum_{i,j} I_x(x+i, y+j)I_y(x+i, y+j) \\ \sum_{i,j} I_x(x+i, y+j)I_y(x+i, y+j) & \sum_{i,j} I_y(x+i, y+j)^2 \end{bmatrix} \quad (2.1)$$

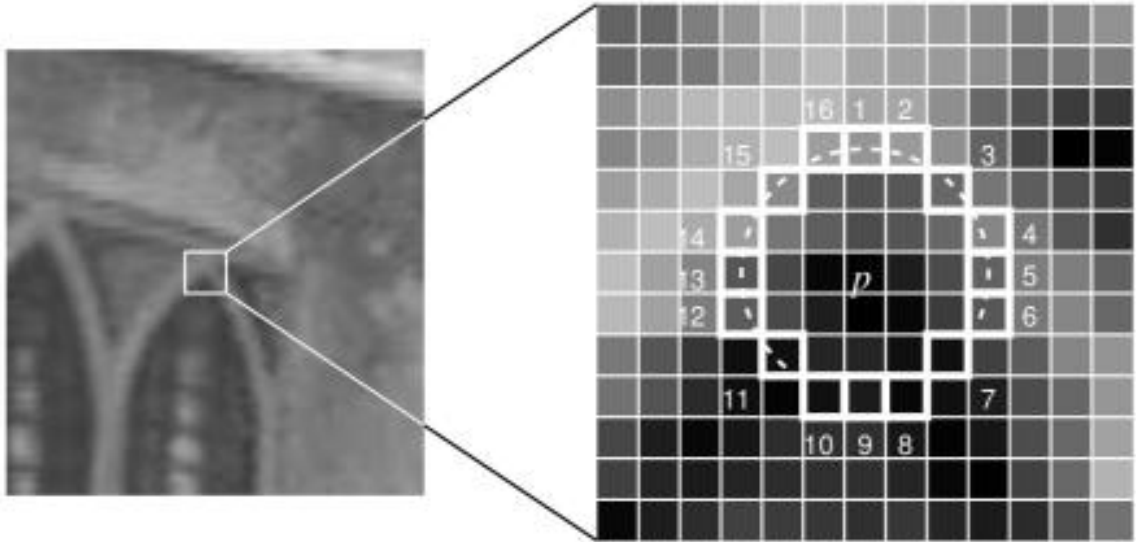
I_x a I_y sú parciálne derivácie I v smeroch x a y . Štruktúra okolia analyzovaného bodu sa určuje pomocou vlastných čísel λ_1, λ_2 z autokorelačnej matice A . Pokiaľ sú tieto vektory veľmi malé, nie je v obraze žiadny príznak detekujúci zakrivenie a oblasť je jednotná. Pokiaľ však je jeden z vektorov väčší, preskúmame túto oblasť. Pokiaľ oba nenulové vektory sú veľké, jedná sa o oblasť hrany. Výpočet vlastných vektorov a vlastných čísel je veľmi náročný a preto Harrisov detektor hrán používa skórovaciu funkciu ρ ktorá je vyjadrená stopami namiesto hodnôt.

$$\rho(x, y) = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(A(x, y)) - k \cdot \text{trace}(A(x, y))^2 \quad (2.2)$$

Zo získaných hodnôt sa prefiltrujú hodnoty ktoré sú nižšie než preddefinovaná hodnota a odstránia sa.

FAST – features from accelerated segment test

Veľmi rýchly algoritmus záujmových bodov predstavil Rosten a Drummond [17]. Tento algoritmus je veľmi vhodný pre real-time spracovanie videa. FAST využíva diskretný kruh okolo kandidátneho bodu. Tento bod je klasifikovaný ako hrana, pokiaľ existuje prilahlý oblúk pixelov dostatočného kontrastu voči stredovému pixelu ktorý tvoria pixely veľkosti minimálne 3/4 z diskretného kruhu. Existuje niekoľko variácií algoritmu FAST, v závislosti od počtu pixelov v oblúku okolo stredového pixelu, napríklad FAST9, FAST12, FAST16. Pre úpravu algoritmu FAST Rosten a Drummond aplikovali strojové učenie na základe rozhodovacieho stromu popisujúceho poradie pixelových oblúkov s cieľom ukončiť testovanie čo najrýchlejšie.



Obr. 2.11: FAST12 algoritmus a detekcia rohov. Prevzaté z [16]

Algoritmus FAST12 využíva okruh skladajúci sa zo 16tich pixelov $s_i, i \in [1..16]$. Hrana je detekovaná vtedy, keď 12 za sebou idúcich pixelov je svetlejších alebo naopak tmavších

ako stredový pixel na základe definovaného prahu d . Pre rýchlejšiu detekciu môžeme najprv detekovať pixely s_1 (vrch), s_9 (spodok), s_3 (vľavo) a s_5 (vpravo). Nakoľko algoritmus FAST neurčuje silu záujmového bodu, skóre ρ je použité na detekovanie svetlých bodov oproti stredovému bodu S^L a tmavších bodov oproti stredovému bodu S^D :

$$\begin{aligned} S^D &= \{s_i(x, y) | s_i(x, y) \leq I(x, y) - d\} \\ S^L &= \{s_i(x, y) | s_i(x, y) \geq I(x, y) + d\} \end{aligned} \quad (2.3)$$

$$\rho(x, y) = \max(\sum_{s_i \in S^D} |s_i - I(x, y)| - d, \sum_{s_i \in S^L} |s_i - I(x, y)| - d)$$

KLT – Kanade-Lucas-Tomasi Tracking

Klasický prístup inkrementálneho sledovania záujmových bodov je prístup od Kanade, Lucasa a Tomasiho (KLT) [11, 20]. Algoritmus pracuje ako tracker, ktorý extrahuje významné body z počiatočného obrazu a potom ich sleduje pomocou optického toku. Sledovanie má za cieľ nájsť parametre zakrivenia, ktoré vznikne ako obraz nasledujúceho snímku na vstupný obraz. Zakrivenie bude často obmedzené na afinnú transformáciu, ktorá je dostatočná na modelovanie deformácie obrazovej časti pozorovanej po malom pohybe kamery. Pri takých malých, prírastkových pohyboch sú afinné transformácie veľmi podobné perspektívnym skreslením, ktoré by však boli oveľa drahšie na výpočet.

Five-Point Algorithm for Essential Matrix

Algoritmus slúži na určenie relatívneho pohybu kamery z 2-bodových korešpondencií podľa Nistéra [15] ktorý vypočíta z piatich bodových korešpondencií. Môžeme vypočítať relatívnu polohu medzi kamerami dekompozíciou použitím SVD³. Optické stredy dvoch uhlov pohľadu kamery musia byť odlišné, v opačnom prípade je triangulácia neurčitá. Na dosiahnutie týchto cieľov musia byť systémy využívajúce SLAM inicializované s odlišným pohybom a neúspešné pokiaľ sa kamera len otáča a nezmení polohu.

2.3 Spôsoby interakcie s 3D objektmi

Jedným z hlavných problémov, ktoré riešime v rozšírenej realite (prípadne aj vo virtuálnej realite alebo obecné v 3D prostredí) je ten, ako s objektmi narábať. Ako mapovať dostupné vstupy na výstupy, objekt rotovať a pohybovať. Podľa niektorých dostupných štúdií [8, 7], kde sa autori venovali práve interakcii virtuálnych objektov na mobilnom zariadení zistíme, že to, čo platilo dakedy už teraz neplatí. Autori článku popisujú výhody využitia počítača oproti mobilu z hľadiska výkonu alebo rozlíšenia a veľkosti obrazovky. V dnešnom svete výkon mobilných zariadení je na veľmi vysokej úrovni a predstavuje plnohodnotnú náhradu počítača v tejto oblasti.

³https://en.wikipedia.org/wiki/Singular-value_decomposition

Interakcia 3D objektov

Pokiaľ sa jedná o manipuláciu objektov napríklad na počítači, potrebujeme mapovať vstupné zariadenia (klávesnicu a myš) na akcie s objektmi v 3D prostredí. Počiatky takejto interakcie môžeme čerpať z článku Hacheta a kolektívu [5], kde sa autori snažili o interakciu 3D a 2D objektov pomocou samostatného zariadenia, ktoré sa dalo ovládať v 6 DoF (six-axis degrees of freedom) viď sekcia 2.2. Toto zariadenie nazvali CAT (Control Action Table), a umožňovalo interaktivitu vo virtuálnom prostredí. Z popisu článku môžeme vyčítať, že toto zariadenie bolo veľmi efektívne a použiteľné v niektorých procesoch.

Vývojom času a technológií sa dostávajú na trh voľne predajné zariadenia, ktoré slúžia na interakciu v 3D prostredí. Jedným zo známych zariadení je aj 6D myš⁴ viď obrázok 2.12, používaná pri rôznych 3D softwaroch CAD.



Obr. 2.12: 6D myš od spoločnosti 3DConnexion prispôbená na prácu so softwérom CAD.

Interakcia 3D objektov v rozšírenej realite

Podľa práce Henryssona a kolektívu z článku [8] navrhli niekoľko spôsobov ovládania objektov v rozšírenej realite pomocou mobilného telefónu, keďže sa však jedná o článok z roku 2005 a technológie boli vtedy na oveľa nižšej úrovni ako sú dnes, nevieme celkom presne posúdiť či sa jedná o relevantné informácie. V tomto článku popisujú interakcie v rozšírenej realite pomocou tlačidlového telefónu, kde interakcie sú v podobe stláčania ovládacích tlačidiel telefónu viď obrázok 2.13. Jedným z možných ovládaní objektov bol aj ten, kde objekt bol neustále v relatívnej pozícii voči mobilnému zariadeniu, tým pádom objekt odrážal presne pohyb mobilného zariadenia v priestore. Rotácia objektov podľa tohto článku je tiež prispôbená tlačidlám mobilného zariadenia a podľa toho objekt rotuje v osách X, Y, Z . Otázkou však ostáva frekvencia pohybu či rotácie objektu, teda ako veľmi a ako často sa objekt po stlačení tlačidla posúva alebo rotuje, prípadne citlivosť tejto interakcie.

⁴<https://www.3dconnexion.cz/products/spacemouse/spacemouse-enterprise.html>



Obr. 2.13: Prvé pokusy manipulácie s 3D objektami v rozšírenej realite pomocou mobilného zariadenia. Na obrázku je ukázaný telefón Nokia 6630 s procesorom 220Mhz. [8]

2.4 ARTable - Pracovisko podporujúce rozšírenú realitu

Význam a využitie ARTable pracoviska

ARTable vznikol pod predstavou tvorby pracovného stolu podporujúceho rozšírenú realitu viď obrázok 2.14. Pojem ARTable vymyslela výskumná skupina Robo@FIT⁵ na vysokej škole Vysoké učení technické v Brně. Dôležitý prínos a predstava významnosti ARTable spočíva v tom, že sa zameriava na malé a stredné podniky, ktoré vykonávajú určité druhy jednoduchých činnosti v manufaktúre, ktoré sa dajú naprogramovať [13]. Jedná sa napríklad o rôznu výrobu a montáž jednotlivých výrobkov. ARTable by mal umožniť jednoducho a efektívne pomôcť zamestnancovi robota naprogramovať a spolupracovať s ním vďaka rozšírenej realite, pričom hlavný prínos a dôraz sa kladie na bezpečnosť pri práci s robotom a jednoduchosť robota naprogramovať a spolupracovať s ním. Prototyp robota má verejne dostupné zdrojové kódy na github⁶.

Technický popis pracoviska

ARTable je systém niekoľkých zariadení skladajúci robota PR2 umiestneného za pracovným stolom (z pohľadu pracovníka) a demonštruje tak princíp kolaboratívneho robota. Pracovného stola označeného markermi, ktoré slúžia na kalibráciu robota a jeho súčastí. Pracovný stôl je obohatený o dotykovú plochu. Na pracovný stôl je premietané rozhranie z projektoru umiestneného na stojane nad stolom. Pracovný stôl slúži ako hlavný zdroj vstupu užívateľského rozhrania s odozvou. Dotyková plocha spolu s projektorom a reproduktormi umiestnenými po bokoch stola poskytuje spätnú väzbu pre užívateľa. Systém využíva 3 Kinecty⁷ umiestnené na statíve z každej strany stola a na hlave robota PR2 pre detekciu objektov umiestnených na stole a v jeho okolí. Systém využíva 3 malé počítače (Intel NUC) pre pripojenie senzorov, projektoru a dotykovej plochy. Následne sú tieto malé počítače pripojené do centrálného počítača. Systém ARTable používa operačný systém ROS, sekcia 2.5.

⁵<http://www.fit.vutbr.cz/research/groups/robo/index.php.en>

⁶<https://github.com/robofit/artable>

⁷<https://en.wikipedia.org/wiki/Kinect>



Obr. 2.14: Pracovisko podporujúce rozšírenú realitu ARTable vyvíjané výskumnou skupinou Robo@FIT na vysokej škole Vysoké učení technické v Brně.

2.5 Robotický operačný systém

ROS je robotický meta-operačný systém (middleware operačného systému) s otvoreným zdrojovým kódom, ktorý poskytuje štandardné služby operačného systému, ako je napríklad abstrakcia hardvéru, kontrola zariadení na nízkej úrovni, implementácia bežne používaných funkcií, prenos správ medzi procesmi a správa balíkov. Je založený na architektúre grafov, kde spracovanie prebieha v uzloch, ktoré môžu prijímať, posilať a multiplexovať senzory, riadenie, stav, plánovanie, pohon a ďalšie. Vo všeobecnosti spája hardvér so softvérom, ktorý umožňuje pokročilé programovacie prostredie na ovládanie hardvéru s nízkou úrovňou.

ROS neposkytuje len bežné služby ako každý operačný systém, ako sú hardvérová abstrakcia, správa procesov a podobne, ale aj synchrónne a asynchrónne volania, centrálnu databázu, konfiguráciu robotov atď. Nie je závislý na programovacom jazyku, môžeme teda využiť programovanie v jazyku C++ alebo aj Python.

Základnou filozofiou operačného systému ROS by sme mohli zhrnúť do týchto 5-tich bodov:

- Peer-To-Peer
- Založené na nástrojoch (mikrokernel)
- Viac-jazyčný
- Úzke prepojenie

- Zdarma a s otvoreným zdrojovým kódom

Vysvetlenie jednotlivých bodov:

Peer-To-Peer: Zložitý robot obsahuje niekoľko palubných počítačov ovládajúcich konkrétny hardvér, rôzne základné dosky a počítače pre zložitejšie výpočty, napríklad spracovanie obrazu a podobne. Peer-to-peer architektúra spojená s vyrovnávacím systémom a vyhľadávacím systémom umožňuje každému komponentu viesť dialóg priamo s inými synchrónnymi alebo asynchrónnymi požiadavkami.

Mikrokernel: Namiesto zložitého runtime prostredia je ROS navrhnutý podľa dizajnu mikrokernelu, ktorý používa množstvo malých nástrojov na prácu s ROS komponentami. Každý príkaz je spustiteľný a výhodou tohoto systému je že problém so spustiteľným príkazom neovplyvňuje ďalšie čo prináša to, že je systém robustnejší a flexibilnejší než centralizované runtime prostredie.

Viac-jazyčný: ROS je jazykovo neutrálny a môže byť naprogramovaný v rôznych jazykoch. Špecifikácia ROS funguje na vrstve správ. Peer-to-peer pripojenia sú aplikované v XML-RPC, ktorý existuje vo veľkom množstve jazykov.

Úzke prepojenie: ROS využíva algoritmy a nástroje v samostatne spustiteľných súboroch. Tým je zaistená maximálna opätovná využiteľnosť a predovšetkým je zachovaná jeho veľkosť. Táto metóda robí ROS ľahko použiteľný. Toto usporiadanie tiež uľahčuje jednotkové testovanie. Nakoniec ROS používa kód (ovládače a algoritmy) z iných projektov s otvoreným zdrojom ako sú napríklad spracovanie obrazu pomocou OpenCV, plánovacie algoritmy z OpenRave a podobne.

Zdarma a s otvoreným zdrojovým kódom: ROS prenáša dáta medzi modulmi pomocou komunikácie medzi procesmi a tak nemusia byť moduly prepojené v rámci jediného procesu, čím je možné využívať rôzne licencie.

Základné pojmy v ROS

Základným princípom ROS je spustenie veľkého počtu menších programov paralelne a zabezpečiť ich komunikáciu medzi sebou či už synchrónne alebo asynchrónne. Napríklad ROS potrebuje dotazovať dáta zo senzorov o určitej frekvencii – dáta z akcelerometra, teplotného senzora, gyroskopu, hĺbkového senzora, kamier a podobne. V ďalšom kroku tieto dáta načítavať, spracovávať a odovzdávať ich algoritmom na vykonanie príslušných operácií, napríklad spracovanie zvuku, zaznamenávanie priestoru a podobne, a nakoniec vykonať pohyb motorov alebo iných súčastí robota. Tento proces sa vykonáva nepretržite a paralelne.

Nižšie popisujem koncept používania ROS nazývaný tiež Výpočtový graf ROS.

Nodes

V systéme ROS je uzol inštanciou spustiteľného súboru. Uzol môže predstavovať algoritmus snímača, motora, spracovania alebo monitorovania atď. Každý uzol, ktorý začne bežať, sa deklaruje uzlu **master**. To sa vráti do architektúry mikrokernelu, pričom každý zdroj je nezávislý uzol.

Master – hlavný uzol

Master je služba, ktorá slúži na deklaráciu a registráciu uzlov, ktorá umožňuje uzlom nájsť ostatné uzly a vymieňať si dáta. Master je implementovaný pomocou XMLRPC ⁸.

Master obsahuje najviac používanú súčasť nazvanú Parameter Server, ktorá je tiež implementovaná pomocou XMLRPC, ktorá slúži ako centralizovaná databáza v ktorej môžu uzly ukladať svoje dáta a poskytovať ich iným službám.

Topics (témy)

Téma je v podstate asynchrónna zbernica správ z uzlov. Správy, ktoré uzly posielajú sa dajú poselať asynchrónne pomocou **topics** alebo synchrónne pomocou **service**. Je to systém na prenos údajov založený na odoberaní a publikovaní správ. Jeden alebo viacero uzlov môže publikovať údaje na **topics** a jeden alebo viacero uzlov môže tieto údaje čítať.

Messages (správy)

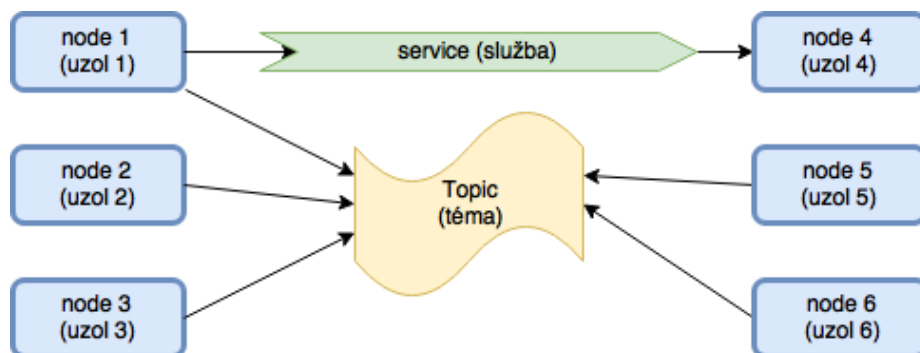
Správa je zložená dátová štruktúra, ktorá obsahuje kombináciu primitívnych typov (znakové reťazce, booleovské čísla, celé a reálne čísla atď). Napríklad uzol predstavujúci robotový servomotor zverejní svoj stav pomocou **topic** so správou obsahujúcou celé číslo reprezentujúce polohu motora, reálne číslo pre svoju teplotu a rýchlosť. Popis správy je uložený v `package_name/msg/myMessageType.msg`. Tento súbor popisuje štruktúru správ.

Services (služby)

Služba narozdiel od **topics**, ktorá je asynchrónna a spĺňa iný druh potreby – slúži na synchrónnu komunikáciu medzi dvoma uzlami. Jej využitie je podobné ako pri diaľkovom volaní. Popis služby je uložený v adresári `package_name/srv/myServiceType.srv`. Tento súbor opisuje štruktúru údajov pre požiadavky uzlov a odpovede.

Bags

Bags sú formáty na ukladanie a prehrávanie údajov správ. Tento mechanizmus umožňuje napríklad zhromažďovanie údajov meraných snímačmi a následné prehrávanie toľkokrát, koľkokrát je potrebné ich simulovať. Je to veľmi užitočné pre ladenie systému.



Obr. 2.15: Diagram znázorňujúci pracovanie ROS.

⁸<https://en.wikipedia.org/wiki/XML-RPC>

2.6 ARKit - Knižnica pre rozšírenú realitu pre iOS

ARKit⁹ je knižnica pre systém iOS ktorú predstavila firma Apple v roku 2017 spolu s príchodom nového operačného systému iOS 11. Táto knižnica slúži pre jednoduchšie programovanie aplikácií v rozšírenej realite pre novšie iPhone a iPady.

ARKit používa vizuálnu inerciálnu odometriu (angl. Visual Inertial Odometry, VIO) na presné sledovanie okolitého sveta vid' kapitola 2.2. VIO spája údaje obraz kamery s údajmi z knižnice CoreMotion. Tieto dva vstupy umožňujú zariadeniu zistiť, ako sa pohybuje v miestnosti s vysokou presnosťou a bez ďalšej kalibrácie.

ARKit beží na procesoroch Apple A9, A10 a A11. Tieto procesory prinášajú prelomový výkon, ktorý umožňuje rýchle pochopenie scény a umožňuje vám vytvoriť podrobný a presvedčivý virtuálny obsah v reálnych scénach. ARKit podporuje aj využitie rendrov v aplikácii ako sú Metal, SceneKit a nástrojov tretích strán, ako napríklad Unity a Unreal Engine.

Pomocou knižnice ARKit na zariadení iPhone a iPad môžete analyzovať scénu zobrazenú v zobrazení kamery a nájsť horizontálne roviny v miestnosti. ARKit dokáže rozpoznať vodorovné roviny ako sú stoly a podlahy a tiež využíva obraz kamery na odhadnutie celkového množstva svetla, ktoré je k dispozícii v scéne a uplatňuje správne množstvo osvetlenia na virtuálne objekty.

Princíp fungovania ARKitu v iOS

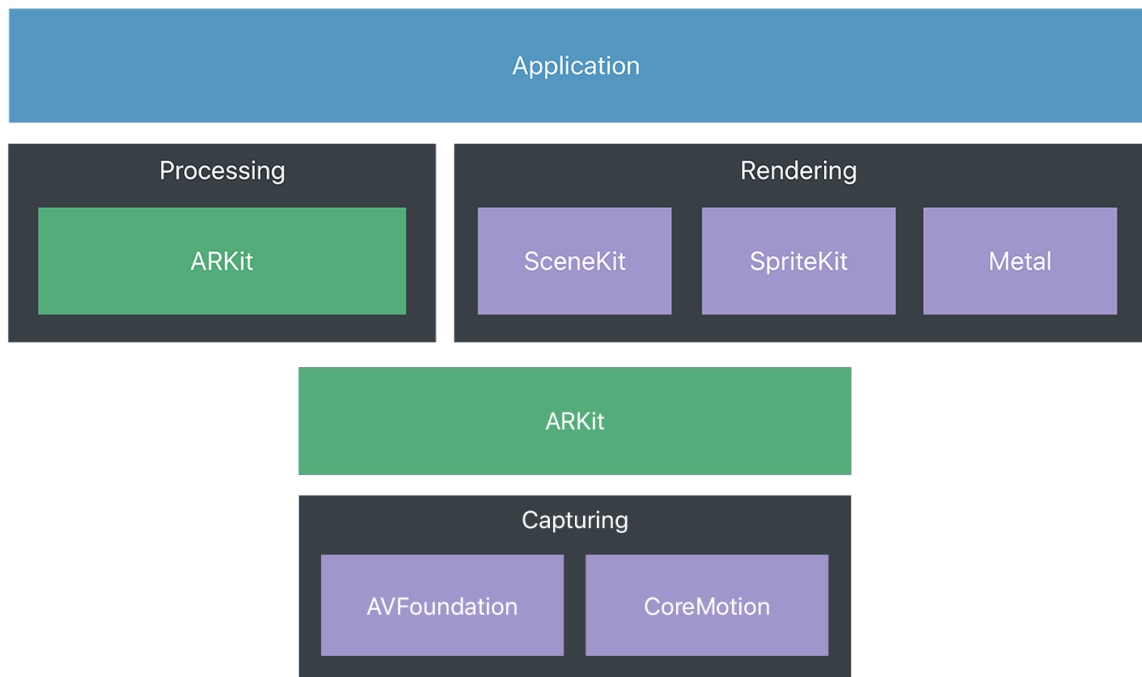
ARKit slúži na spracovávanie vstupných dát dostupných v knižniciach `AVFoundation` a `CoreMotion` znázornené na obrázku 2.16, podľa WWDC konferencie [1]. `AVFoundation` poskytuje obrazové dáta z kamery zariadenia a parametre kamery ako je napríklad ohnisková vzdialenosť, clona kamery a podobne. K tomu, aby malo zariadenie prehľad o pohybe zariadenia v priestore ARKitu poskytuje dáta knižnica `CoreMotion`, kde sú k dispozícii údaje o rotácii a zmene rotácie zariadenia.

K vykreslovaniu scény, ktorú máme k dispozícii môže využiť ARKit knižnice ako sú `SceneView`, ktorá poskytuje možnosti vykreslovania 3D objektov, `SpriteKit`, ktorý slúži na prácu s 2D objektmi a `Metal`, ktorý umožňuje vytvoriť vlastný render.

Objekt, ktorý v relácii ARKitu koordinuje hlavné procesy sa nazýva `ARSession`. Medzi tieto procesy patrí už spomínané čítanie údajov z hardvéru snímača pohybu zariadenia, ovládanie vstupného fotoaparátu a vykonávanie analýzy obrazu na zachytávaných záberoch fotoaparátu. Referencia tohto objektu syntetizuje všetky tieto výsledky na vytvorenie korešpondencie medzi skutočným priestorom, kde sa zariadenie nachádza a virtuálnym priestorom modelovaným zariadením.

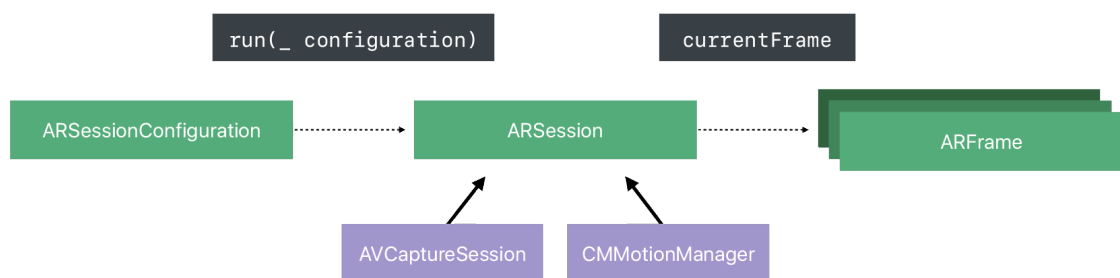
Spustenie relácie vyžaduje konfiguráciu. Podtriedy abstraktnej triedy `ARConfiguration` určujú, ako ARKit sleduje polohu a pohyb zariadenia vzhľadom na skutočný svet a tým ovplyvňujú druhy nastavenia konfigurácie, ktoré môžeme vytvoriť. Napríklad používanie `ARWorldTrackingConfiguration` konfigurácie využíva zadnú kameru zariadenia, môžeme využiť hit-testing, hľadanie horizontálnej plochy a poskytuje plné využitie ARKitu. Zároveň však môžeme overiť, či dané zariadenie podporuje danú konfiguráciu.

⁹<https://developer.apple.com/arkit/>



Obr. 2.16: Princíp pracovania ARKitu v iOS aplikácii. ARKit slúži na spracovávanie vstupných dát z kamery a pohybových snímačov.^[1]

Podľa obrázka 2.17 sa pomocou `ARSessionConfiguration` spúšťa `ARSession`. Výstup, ktorý poskytuje `ARSession` je v podobe objektu `ARFrame`, ktorý poskytuje v zaznamenaných snímkoch náhľad snímku, časové razítko a hĺbkové dáta. Čo sa týka scény, v tomto objekte máme prístup ku kamere, teda jej polohu voči počiatočnému súradnicovému systému, rotáciu, nastavenia a taktiež odhad zdroja svetla. `ARFrame` tiež umožňuje vykonať hit test, čo je v podstate hľadanie objektov v pohľade lúča z kamery do priestoru na základe rotácie zariadenia. Pomocou hit testov môžeme zistiť, či užívateľ mieri na daný objekt alebo aké objekty sa nechádzajú v lúči.



Obr. 2.17: Objekt `ARSession` sa spúšťa pomocou konfigurácie a jeho výstupom sú `ARFrames`.

Odporúčania užívateľského rozhrania pre ARKit od Apple

Apple s predstavením novej knižnice ARKit poskytol aj odporúčania na rozhranie pre prácu s rozšírenou realitou ¹⁰. Keďže rozšírená realita predstavuje zobrazenie virtuálnych objektov nad vrstvou zobrazenia reálneho sveta na obrazovke displeja, vytvára tak ilúziu že dané objekty existujú. Tieto odporúčania vedú k správnej interakcii s virtuálnymi objektami. Niektoré druhy odporúčaní, ktoré som považoval za dôležité:

Zážitok z dizajnu

- **Využitie celého displeja.** Pohodlnejšie pre užívateľov bude, keď využijeme toľko miesta, koľko máme k dispozícii pri interakcii s virtuálnymi objektmi.
- **Predpokladajme, že ľudia budú používať aplikáciu v prostrediach, ktoré nie sú optimálne pre AR.** Ľudia môžu otvoriť aplikáciu na mieste, kde nie je veľa priestoru na pohyb, alebo nie sú veľké ploché plochy. Snažme sa predvídať scenáre, na ktoré bude aplikácia využívaná.
- **Využívať haptickú a zvukovú spätnú väzbu.** Zvukový efekt alebo pocit nárazu je skvelý spôsob, ako potvrdiť, že virtuálny objekt sa dostal do kontaktu s fyzickým povrchom alebo iným virtuálnym objektom.
- **Vždy, pokiaľ je to možné, poskytovať rady v kontexte.** V praxi to znamená to, že umiestnenie trojrozmerného ukazovateľa okolo objektu je omnoho viac intuitívnejšie ako zobrazovať len tlačidlo s textom.

Vstup do rozšírenej reality

- **Je potrebné užívateľa informovať o tom, že inicializácia priestoru prebehla.** Inicializačný proces, počas ktorého sa vyhodnocuje okolie začína vždy, keď sa aplikácia s rozšírenou realitou zapne. Môže trvať až niekoľko sekúnd. Na to, aby sme predišli zbytočnému zmätku používateľa a urýchlili inicializačný proces treba informovať užívateľa a povzbudzovať ho v preskúmaní okolia.

Vkladanie virtuálnych objektov

- **Pomôcť pohopiť ako detekovať plochu a vložiť objekt.** Vizualný indikátor, je skvelý spôsob ako pokázať na to, že plocha bola detekovaná a umožňuje na ňu vložiť objekt. Pri umiestňovaní objektu by sa mal indikátor meniť podľa toho, či je možné vložiť objekt na danú pozíciu alebo nie.
- **Odpovedať na užívateľove umiestnenie objektov okamžite.** Tým, že skúmanie a mapovanie priestoru prebieha postupne nejaký čas, je potrebné umiestniť objekt ihneď, keď to užívateľ zadá a následne opatrne a pomaly po presnejšej detekcii plochy tento objekt zarovnávať. Napríklad ak sa objekt umiestni pod hladinou plochy a treba ho dostať práve na detekovanú plochu.
- **Vyhnuť sa snahe zarovnávať všetky objekty.** Keďže detekcia plochy prebieha neustále a neustále sa mení jej presnosť, netreba vo vysokej miere dbať na úplne presné zarovnanie objektov podľa detekovanej plochy a jej pozície.

¹⁰<https://developer.apple.com/ios/human-interface-guidelines/technologies/augmented-reality/>

Interakcia s virtuálnymi objektami

- **Priama manipulácia s objektami.** Je viac intuitívne manipulovať s objektom priamo na obrazovke a predstavovať tak priamy dotyk s objektom, ako užívateľovi poskytnúť ovládacie prvky mimo objektu na kraji obrazovky a spôsobiť mu zmätok. Treba však mať na pamäti to, že nie vždy je toto ovládanie lepšie a pohodlnejšie, napríklad pri ovládaní letiaceho lietadla.
- **Použitie štandardných gest.** Nie vždy je rozumné vymýšľať nové spôsoby interakcie a sťažovať tak užívateľovi manipuláciu s objektami. Štandardné gestá ako napríklad rotovanie objektu dvoma prstami a jeho posúvanie jedným prstom je zaužívané vo viacerých aplikáciách a užívateľovi bude tak od začiatku všetko jasné.
- **Snaha udržať jednoduché interakcie.** Nie je vždy jednoduché manipulovať s objektami ktoré sú v 3D priestore pomocou dotykov, čo predstavuje 2D interakciu. Odporúča sa napríklad obmedziť rotovanie objektu pozdĺž jeho stredovej osi, alebo pohybovať s objektom len v 2 smeroch po povrchu.
- **Reagovať na gestá s primeraným odhadom v blízkosti virtuálnych objektov.** Niekedy je obtiažne manipulovať s objektmi pokiaľ sú veľmi malé, tenké alebo umiestnené ďalej. Treba v aplikácii zisťovať či v blízkosti dotyku existuje objekt a reagovať na toto gesto interakciou s daným objektom. Je teda lepšie predpokladať, že užívateľ chce s daným objektom pracovať.
- **Udržať pohyby objektov hladké.** Objekty by nemali trhane meniť svoje miesto, pohybovať sa v priestore sekavým pohybom a rôzne skákať z miesta na miesto. Ich pohyby by mali byť hladké a plynulé.
- **Vyskúšať ďalšie metódy interakcie.** Gestá nemusia predstavovať jediný spôsob ako s objektami narábať. Môžeme využiť napríklad aj výpočet vzdialenosti predmetu od kamery, približovanie sa k objektu alebo polohu kamery voči objektu. Napríklad využiť polohu telefónu voči postave v hre, ktorá priblížením telefónu k postave obráti hlavu k užívateľovi a podobne.

Zvládanie problémov

- **Umožniť užívateľovi resetovať prostredie v ktorom pracuje.** Ak užívateľ zistí, že prebehla zlá detekcia prostredia alebo má pocit že niečo nieje v poriadku, musí mu byť umožnené toto detekované prostredie resetovať a začať tak od začiatku s detekciou prostredia.
- **Navrhnuť možné opravy, ak sa vyskytnú problémy.** Analýza používateľského prostredia a detekcia povrchu môže zlyhať alebo trvať príliš dlho z rôznych dôvodov - nedostatočné svetlo, nadmerne reflexný povrch, povrch bez dostatočných detailov alebo príliš rýchly pohyb kamery. Ak je aplikácia informovaná o týchto problémoch, mali by sme ponúknuť návrhy na ich riešenie – upozorniť užívateľa na zlé svetlo v okolí, nedostatočnú detekciu prostredia, rýchly pohyb kamery a pod.

V tejto práci sa mimo iné veci zaujímam aj práve návrhom rozhrania pre manipuláciu virtuálnych objektov rozšírenej realite a pokúšam sa korešpondovať s odporúčaniami od spoločnosti Apple spolu s vlastnými návrhmi.

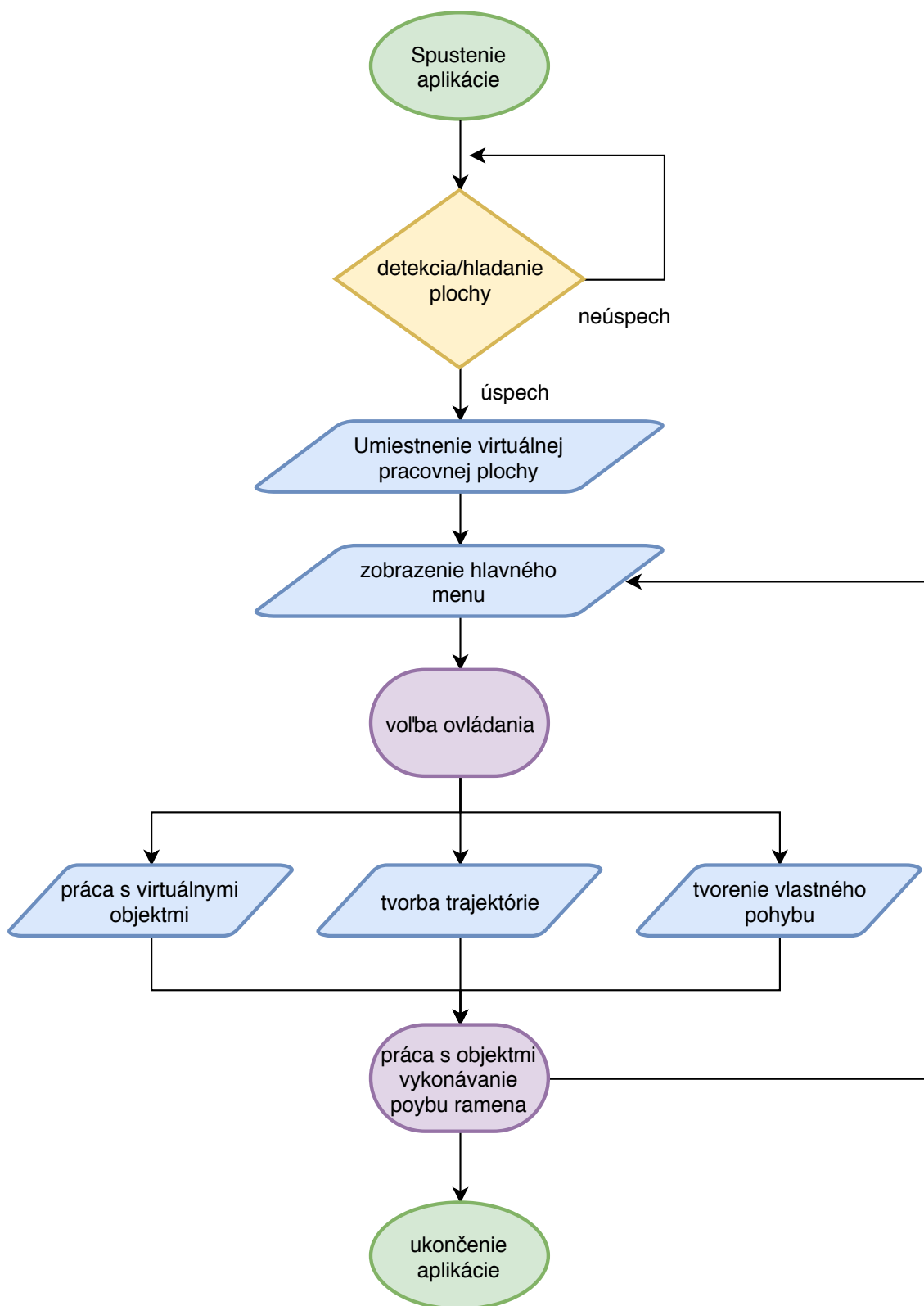
Kapitola 3

Návrh riešenia aplikácie pre ovládanie robotického ramena

V kapitole 2 uvádzam, že rozšírená realita má mnoho druhov využitia a množstvo rôznych riešení. Musí však spĺňať dané požiadavky a hlavne schopnosť pracovať v reálnom čase. Hlavným cieľom diplomovej práce bolo navrhnuť užívateľské rozhranie pre aplikáciu na ovládanie robotického ramena s aplikovaním rôznych metód ovládania virtuálnych objektov v rozšírenej realite. Následne na základe užívateľského testovania určiť, ktoré druhy ovládania sa užívateľom zdajú intuitívne, s ktorými vedia pracovať a ktorým častiam aplikácie nerozumejú.

Zobrazovacie zariadenie môže byť mobilný telefón s uhlopriečkou do 6", alebo tablet, kde obrazovky siahajú až do veľkosti 13". Pokiaľ chceme, aby užívateľ disponoval s čo najväčším pracovným priestorom, mal čo najlepšie možnosti zobrazovania ovládacích prvkov a čo najpohodľnejšie sa pohyboval so zariadením v priestore pričom by nestratil prehľad o scéne, môžeme považovať mobil ako zobrazovacie zariadenie za nepoužiteľné vzhľadom na množstvo UI prvkov ktoré sa budú na obrazovke nachádzať.

Aplikácia na ovládanie robotického ramena by mala spĺňať najmä príjemne užívateľské rozhranie, ktoré bude v súlade napríklad s odporúčaniami užívateľského prostredia od firmy Apple popisované v sekcii 2.6. Aplikácia bude podporovať viac módov práce s objektmi a tvorenie pokynov pre robotické rameno. V tejto kapitole som navrhol niekoľko spôsobov ovládania robotického ramena a dať užívateľovi možnosť sa rozhodnúť, ako bude chcieť s robotickým ramenom pracovať. Na znázornenom diagrame 3.1 demonštrujem, ako by mala výsledná aplikácia pracovať.



Obr. 3.1: Diagram priebeh činností v aplikácii.

3.1 Typické využitie a prínos

Typickým využitím mojej aplikácie si môžeme predstaviť využitie v menších podnikoch, kde pracovníci vykonávajú opakované činnosti, ktoré by sa dali automatizovať a naprogramovať. Ako príklad môžem uviesť nanášanie lepidla po obvodoch nejakého predmetu, napríklad kovového rámu, alebo princíp využitia robota ako svojho pomocníka pri práci s človekom ako je spomínané v sekcii 2.4. Pri interaktivite robota s človekom musíme dbať na bezpečnosť, aby nedošlo k nehodám medzi robotom a človekom, ale aj na presnosť niektorých operácií, presnosť ovládania robotického ramena a naprogramovanie úkonov s tým spojených.

Keďže mnohokrát nám rozšírená realita môže priniesť nové pohľady a dopĺňať tak naše reálne prostredie, vieme nájsť využitie aj v priemysle práve pri ovládaní robotického ramena. Nemusí to byť nutne ovládanie objektov s ktorými bude robotické rameno pracovať, môžeme si predstaviť aj zakázané zóny, kde rameno nebude mať dosah, zóny, kde bude musieť rameno pracovať nižšou rýchlosťou alebo väčšou presnosťou.

Ďalším z možných spôsobov využitia môže byť pracovanie vo virtuálnom priestore, ktoré odráža reálne pracovisko robotického ramena vzdialeného od tohto priestoru. Pracovník bude môcť ovládať robotické rameno mimo reálneho pracoviska, vytvárať scénu a plánovať pohyby objektov napríklad v komfortnejšom prostredí, prípadne demonštrovať tak prácu ramena bez potreby použitia reálneho pracoviska.

Rozšírená realita sa stáva súčasťou každodenného života a je predpoklad, že jej prínos v priemysle sa odstupom času stane veľmi významný. Či už ide o simulovanie rôznych úkonov, ktoré robotické rameno musí vykonať alebo plánovanie týchto úkonov pre výrobnú halu, malo by to dopomôcť k zvýšeniu efektivity práce.

Časté problémy

Podľa článkov od Henrysson a spol. [8, 7], najčastejšie riešime problémy ohľadom citlivosti či už translácie objektov alebo rotácie. Tieto časté úkony si vyžadujú veľkú pozornosť pri zmene polohy či rotácie objektu nakoľko sa v priemysle často rieši presnosť. Tento dôležitý faktor tiež do istej miery ovplyvňuje spôsoby manipulácie objektov. Pri citlivosti ovládania riešime aj to, o akú vzdialenosť alebo uhol sa má objekt rotovať a hýbať vzhľadom na interakciu užívateľa. Pokiaľ robíme s objektmi o veľkosti niekoľko centimetrov, nemôžeme mapovať pohyb zariadenia v metroch a naopak. Základnou mierkou, ktorá sa používa v rozšírenej realite je mierka v metroch. Keďže však robíme v menšom priestore s menšími objektmi ako meter, navrhol som používať centimetrovú mierku pre transláciu objektov, ktorá bude dostačujúca. V rotácii budú postačovať stupne s celočíselným zaokrúhľaním. Môžeme však uvažovať o ručitom prichytení a nastavení rozostupu uhlov pri rotovaní napríklad zmena vždy o 5 stupňov.

3.2 Požiadavky na systém

Práca sa sústreďí na vytvorenie systému pre manipuláciu s objektmi vytvorenými v rozšírenej realite pomocou tabletu. Či už ide o objekt typu kocka, guľa, ihlan alebo úsečka orientovaná v 3D prostredí, každý z týchto objektov má svoje vlastnosti ako vlastné vnútorné parametre a relatívne parametre vzhľadom na scénu v rozšírenej realite.

Požiadavkou systému a jeho kľúčovou vlastnosťou je určenie (alebo nájdenie) vhodného spôsobu, ako s objektom pracovať v rozšírenej realite. Vieme teda, že systém na prácu s objektom by mal mať nasledovné vlastnosti:

- Vkládanie nového objektu
- Translácia objektu
- Rotácia objektu
- Mazanie objektu

Vkládanie nového objektu

Tento úkon si vyžaduje otázku, kam a ako objekt vkladať. Jeden z možných spôsobov je riadiť pozíciou kamery vo vytvorenom virtuálnom priestore a objekt vkladať priamo pred kameru, prípadne do nejakej vopred určenej pozície. Pokiaľ však ide o manipuláciu reálnych objektov robotickým ramenom, predpokladám, že dané rameno má obmedzený dosah a pracuje vo vlastnej „pracovnej ploche“, teda vo vlastnom vymedzenom priestore ktorým disponuje. Nechcem virtuálne objekty vkladať rôzne do prostredia a následne im nastavovať korektnú pozíciu. Ideálne je objekt vkladať priamo na pracovnú plochu. Navrhujem preto v systéme využiť „virtuálnu pracovnú plochu“, ktorá bude predstavovať plochu umiestnenú na začiatku spustenia aplikácie a v tejto ploche sa bude s objektmi pracovať.

Translácia a rotácia

Po vložení objektu chcem v systéme umožniť ovládanie pozície a rotácie objektu tak, aby užívateľ napríklad nestratil prehľad o tom, kde sa objekt nachádza, minimalizovať užívateľské vstupy a v neposlednom rade dbať na to, aby si vlastnou interakciou po displeji tabletu/telefónu nebránil v prehliadaní scény na obrazovke. Navrhujem vytvorenie ovládacích prvkov pre ovládanie osí X, Y, Z , pomocou ktorých bude užívateľ nastavovať či sa má objekt hýbať alebo nie.

Mazanie objektov

Mazanie vytvorených objektov nie je príliš častá operácia a preto stačí len pri označení objektu ponúknuť užívateľovi objekt zmazať. Mazanie objektu by malo užívateľovi umožniť vrátiť tento krok späť, prípadne ešte pred samotným vymazaním objektu užívateľa vyzvať na potvrdenie tejto akcie.

Ďalšie vlastnosti systému

Práca s objektmi nebude jediný mód, ako ovládať robotické rameno v tomto systéme. Navrhol som do neho pridať možnosť vytvárania trajektórií vedúcich napríklad od objektu, k objektu, alebo tvorbu trajektórie po určitej ploche. Toto riešenie umožní vytvoriť pohyb ramena za určitým objektom, alebo napríklad „pritlačiť“ jeden objekt k druhému. Viac o tomto móde v sekcii 3.3.

Neoddeliteľnou súčasťou aplikácie bude jej prepojenie s robotickým ramenom. Keďže robotické rameno má vlastný suradnicový systém vo svojej pracovnej ploche a aplikácia pracuje vo vlastnom súradnicovom systéme, bude potrebné tieto dva súradnicové systémy zlúčiť pomocou transformačnej matice. Navrhol som zmeniť súradnicový systém na zariadení v rozšírenej realite a prispôbiť sa tak súradnicovému systému robotického ramena.

Táto zmena súradnicového systému bude vykonaná po umiestnení virtuálnej pracovnej plochy, kde plochu budem umiestňovať tak, aby zapadla svojim súradnicovým systémom do súradnicového systému robotického ramena. Navrhol som, že počas bežania aplikácie sa bude môcť užívateľ pripojiť k robotickému ramenu a aplikácia bude následne automaticky odosielať súradnice objektu pri jeho uchopení a pustení. Nie je potrebné odosielať celú trajektóriu súradníc objektu počas manipulácie s ním, v tomto prípade robotické rameno vykoná len naplánovaný pohyb z bodu A do bodu B.

Užívateľ by mal vedieť či je komunikácia s robotickým ramenom úspešná alebo nie. Navrhujem preto v rozhraní, kde sa užívateľ bude pripájať na robotické rameno pomocou IP adresy a portu aj jednoduché logovanie odosielaných/príjimaných dát. Viac o tejto komunikácii popisujem v sekcii 3.6.

Pracovná plocha

Ako som spomínal v prechádzajúcich častiach textu, dôležitou súčasťou systému je aj pracovná plocha. Plocha, ktorá bude informovať užívateľa v akom priestore môže pracovať, aký prístor a dosah má robotické rameno a aká je pozícia objektov voči tejto pracovnej ploche.

Počas detekcie plochy je dôležité užívateľa informovať čo sa deje a aké kroky aplikácia vykonáva. Toto navrhujem v aplikácii značiť ako textom, tak aj intuitívne farebné vyznačenie detekovanej plochy. Pri inicializácii prostredia navrhujem zobrazovať červenú mriežku, ktorá bude predstavovať veľkosť detekovanej plochy. Po tom, ako bude plocha dostatočne veľká – minimálna veľkosť by mala byť taká, aká veľká je pracovná plocha ARTable, mriežka zmení farbu na zelenú a aplikácia užívateľa informuje o tom, že môže umiestniť „virtuálnu pracovnú plochu“, teda plochu na ktorej bude pracovať a kde bude mať robotické rameno dosah.

3.3 Módy systému

Na základe systémových požiadaviek na systém a predstavou ako s robotickým ramenom narábať a aké úkony môže užívateľ využívať som navrhol v aplikácii využitie štyroch rôznych módov. Prvý mód bude na pridávanie, mazanie a manipuláciu s virtuálnymi objektmi. Ďalší z módov bude tvorenie a editácia pomocných rovin, ktoré budú potrebné pre ďalšie 2 módy v ktorých sa bude tvoriť trajektória pohybu robotického ramena na základe bodov spojených priamkami na pomocných rovinách alebo nahrávanie voľnej trajektórie. Tieto módy ovládania popíšem podrobnejšie v tejto sekcii.

Objekty

Mód, ktorý tvorí jadro aplikácie pre pridávanie, mazanie a manipuláciu objektov. Pridávanie objektov prebieha tak, že užívateľ stlačí tlačidlo plus na ľavej strane obrazovky a následne sa zobrazí ponuka dostupných objektov ktoré užívateľ môže vytvoriť. Užívateľ môže pridať virtuálne objekty ako sú guľa, kocka, valec, ihlan a kužeľ. Všetky tieto objekty budú mať defaultnú veľkosť. Po kliknutí na obrázok objektu sa objekt vytvorí s počiatočnými súradnicami 0,0,0. Mazanie objektu je navrhnuté tak, že užívateľ musí mieriť na objekt terčíkom umiestneným v strede obrazovky tak, aby bol objekt označený a následne môže užívateľ zmazať objekt postranným tlačidlom na ktoré musí kliknúť 2x. Označovanie objektov teda prebieha automaticky v tomto móde tým, že užívateľ mieri na daný objekt terčíkom. To, že je objekt označený znázorňujú farebné osi X , Y , Z objektu ktoré sa zobra-

zia po jeho označení. Metódy akými môžeme vytvorené objekty ovládať sú popísané v sekcii 3.4, avšak každé ovládanie objektu prebieha tak, že objekt s ktorým chceme manipulovať musí byť označený a následne postranným ťahacím tlačidlom umiestneným v pravej časti obrazovky môže užívateľ objekt „chytiť“ a meniť jeho polohu a rotáciu dostupnými metódami a následne „pustiť“. Týmto ťahacím tlačidlom si vie užívateľ prispôbiť citlivosť počas narábania s objektom ťahaním dohora (napríklad pre $1.5x$ zrýchlenie) alebo do dola (napríklad pre $0.5x$ spomalenie).

Pomocné roviny

Pomocné roviny budú slúžiť na vytváranie trajektorie robotického ramena. Navrhol som využitie týchto pomocných rovín pre to, aby užívateľ jasne vedel kam a ako môže ukladať jednotlivé body trajektórií tým a pre jednoduchosť ukladania bodov pod určitým uhlom. Pomocné roviny sa budú pridávať do pracovnej plochy podobne ako objekty. Užívateľovi sa zobrazí ponuka rovín, ktoré môže vytvoriť s vopred definovanou veľkosťou, napríklad $10 \times 10\text{cm}$, alebo $10 \times 15\text{cm}$. Manipulácia s rovinami bude taká istá ako manipulácia s vytvorenými objektmi a pre jednoduchšie rozoznanie pomocných rovín budú farebne odlišné od virtuálnych objektov a mierne priehľadné. Vďaka pomocným rovinám užívateľ bude môcť vyskladať rôzne zakrivenú plochu, na ktorú následne bude môcť ukladať body trajektórie.

Trajektória priamkami

Pokiaľ užívateľ pridal do priestoru pomocné roviny, môže na ne pridávať body trajektórie pohybu. Pre pridávanie bodov využijeme terčík umiestnený v strede obrazovky, ktorý bude indikovať možnosť pridania bodov tým, že jeho farba bude zelená, v opačnom prípade červená. Užívateľ bude postupne vkladať body na vytvorené pomocné roviny a následne sa body budú spájať priamkami. Užívateľ bude mať možnosť túto trajektóriu aj editovať a to stlačením tlačidla editácie umiestneným vľavo. Po stlačení tlačidla editácie sa umiestnené body na rovinu zväčšia a zvýrazia a užívateľ s nimi bude môcť manipulovať ako s objektmi, teda meniť ich pozíciu, prípadne jednotlivé body zmazať.

Trajektória voľným pohybom

Nahrávanie voľného pohybu je veľmi podobné vytvoreniu virtuálnej cesty a vkladá body na pomocnú rovinu súčasne s pohybom terčíka po ploche. Užívateľovi pri mierení terčíkom na pomocnú rovinu je indikované, že môže na plochu umiestňovať body trajektórie. Umiestňovanie bodov bude prebiehať stlačením a držaním tlačidla vpravo a následným pohybom terčíka po rovine. Takto užívateľ vloží množstvo bodov naraz a v niektorých prípadoch to môže byť rýchlejšie ako pri vkladaní samostatných bodov.

3.4 Metódy manipulácie s virtuálnymi objektmi

Keďže táto práca je zameraná najmä na manipuláciu s virtuálnymi objektmi, navrhoval som rôzne spôsoby translácie či rotovania objektov, ktoré som následne testoval aby som sa uistil, či je daná metóda použiteľná. Tieto metódy sú navrhnuté zväčša na základe vlastnej intuitivity a podľa článku [8], v ktorom autori riešia priamu manipuláciu objektov na základe zmeny polohy zariadenia v priestore a rozhodol som sa túto metódu otestovať. Každá z nižšie uvedených metód slúžiacich na manipuláciu objektov bude umožňovať fixovať transláciu alebo rotáciu pre osi X, Y, Z , vďaka čomu môžeme predísť zmenám v nevyžiadanom smere

osí. Pri každom počiatku rotovania alebo translácie sa uloží počiatočná pozícia a rotácia zariadenia do premenných `sPosition` a `sRotation` definovaných podľa štruktúr znázornených vo výpise kódu 3.1.

```
1 struct StartPosition {
2     var x : Float?
3     var y : Float?
4     var z : Float?
5 }
6
7 struct StartRotation {
8     var xAngle : Float?
9     var yAngle : Float?
10    var zAngle : Float?
11 }
```

Kód 3.1: Štruktúra uloženia počiatočnej pozície a rotácie zariadenia pri manipulácii objektov.

Translácia – Priama manipulácia

Pod pojmom manipulácia objektu si môžeme predstaviť ovládanie objektu na základe zmeny polohy zariadenia v priestore. Táto metóda demonštruje spôsob manipulácie s objektom systémom „Drag&Drop“. Pokiaľ sa poloha zariadenia zmení v určitom smere v 3D priestore a zariadenie sa pohne z bodu *A* do bodu *B* napríklad o 10 cm, objekt sa taktiež posunie týmto smerom o 10 cm v závislosti od povolených osí v ktorých sa objekt môže pohybovať. Veľkou výhodou tohto ovládania je, že virtuálny objekt priamo kopíruje pohyby zariadenia a užívateľ má lepšiu predstavivosť ako s objektom manipulovať. Pseudokód metódy je popísaný vo výpise kódu 3.2.

```
1 selectedObject.position.x += sPosition.x - cameraPosition.x
2 selectedObject.position.y += sPosition.y - cameraPosition.y
3 selectedObject.position.z += sPosition.z - cameraPosition.z
```

Kód 3.2: Spôsob priamej manipulácie objektov pomocou cez osi *X, Y, Z* s využitím počiatočnej a aktuálnej pozície zariadenia.

Translácia – Tiahlo

Ďalší z možných spôsobov manipulácie s objektom je označený (aktívny) objekt „posúvať“ po jednej z osí *X, Y, Z*. Translácia objektu prebieha pomocou ovládacieho prvku na zmenu citlivosti, kde namiesto citlivosti sa bude objekt pohybovať v zápornom alebo kladnom smere o určitú časť v cm. Tento spôsob ovládania vie objekt posúvať len o maximálne niekoľko centimetrov, následne musí používateľ prerušiť posúvanie a začať odznova. Tento spôsob ovládania je pomerne veľmi jednoduchý a intuitívny avšak veľmi pomalý pri presúvaní objektov vo väčšej vzdialenosti. Jeho veľkou výhodou je to, že užívateľ je schopný presúvať objekt veľmi presne s citlivosťou posúvania na milimetre. Pseudokód použitia metódy je znázornený vo výpise kódu 3.3.

```

1 // sensitivityOption can be in <-2 cm, 2 cm>
2 selectedObject.position.x = sOriginalPosition.x +
    sensitivityOption

```

Kód 3.3: Spôsob priamej manipulácie objektov pomocou cez osi X, Y, Z s využitím počiatočnej a aktuálnej pozície zariadenia.

Rotácia – Podľa rotácie zariadenia

Tento spôsob rotácie objektu odráža rotovanie zariadenia a znova si pri ňom môže užívateľ zvoliť cez ktoré osi bude objekt rotovať. Po stlačení ovládacieho prvku na ovládanie citlivosti sa uloží počiatočná pozícia rotácie zariadenia a následne sa zmena rotácie zariadenia aplikuje priamo na objekt taktiež s využitím citlivosti. Pokiaľ však chceme rotovať objekt o napríklad viac ako 40° , stane sa, že objekt zmizne z displeja a užívateľ nevidí či objekt rotuje správne. Pri manipulácii s objektmi by sme vždy mali mať prehľad o tom, či sa požadovaná zmena vykonáva alebo nie. Táto metóda je vhodnejšia na presnejšie rotovanie objektu, kde pri používaní veľkých uhlov nemá veľmi využitie. Riešením tohto problému je, že užívateľ bude prerušovať a znovu začínať rotáciu objektu nanovo aby mal stále objekt v dohľade.

Rotácia – Podľa uhlu zovretého pohybom zariadenia

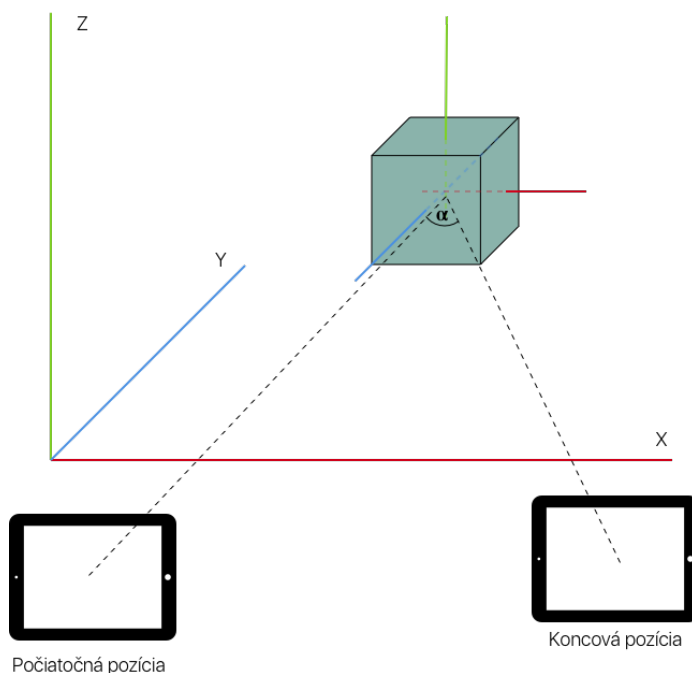
Táto metóda rotovania objektu funguje na základe uhlu počiatočnej a koncovkej pozície zariadenia pri rotovaní objektu okolo svojej osi, teda aký uhol zoviera zmena pozície zariadenia. Pre výpočet uhla o ktorý má objekt rotovať je použitá kosínusová veta odvodená zo základného tvaru $a^2 = b^2 + c^2 - 2bc \cdot \cos \alpha$. Veľkosti úsečiek a, b, c vypočítame jednoducho pomocou rozdielov pozícií objektu, počiatočnej a koncovkej pozícií zariadenia. Princíp metódy je načrtnutý na obrázku 3.2. Táto veta sa dá použiť pri všetkých troch smeroch rotovania objektu a užívateľ nestráca prehľad o tom, kde sa objekt nachádza, pretože objekt sa v ňom vždy smerom k zariadeniu.

Zmena veľkosti objektu

Metóda na zmenu veľkosti objektu nie je často využiteľná no aj napriek tomu by som ju rád dal do návrhu. Keďže objekty ktoré vytvárame majú nejakú predvolenú veľkosť, umožním tak užívateľovi meniť veľkosť tohto objektu zmenou merítka. Pre zmenu veľkosti bude musieť užívateľ objekt označiť a následne ovládacím prvkom na zmenu citlivosti bude môcť objekt zmenšiť alebo zväčšiť v nejakom pomerovom rozsahu.

Zmena citlivosti

Niektoré operácie si vyžadujú prácu s rôznou citlivosťou. Jedná sa teda o zrýchlený pohyb objektov alebo naopak, spomalený. Tým, že pracujeme v rozšírenej realite a potrebujeme s objektmi narábať rôzne a nejde to vždy v mierke 1 : 1 (pohyb zariadenia odráža presne pohyb objektu). Citlivosť pohybu objektu v priestore môže určovať rýchlosť a zároveň presnosť pohybu po osiach X, Y, Z . Citlivosť rotácie nám určuje, v akom pomere objekt rotovať vzhľadom na pohyb zariadenia v priestore. V nastavovaní citlivosti môžeme rotovať objekt napríklad priamo po určitom stupňovom rozsahu, napríklad vždy o 5° .



Obr. 3.2: Jedna z metód rotovania objektu pomocou kosínusovej vety na základe zmeny polohy zariadenia a zovretia uhla.

3.5 Navrhnuté grafické rozhranie

Už pri návrhu GUI bolo nutné vyriešiť problémy rozmiestnenia ovládacích prvkov aplikácie na základe toho, ako často užívateľ s jednotlivými ovládacími prvkami pracuje, alebo toho, ako tablet drží a bude s ním manipulovať. Navrhol som grafické rozhranie znázornené na obrázku 3.3 na základe potrieb užívateľa. Toto grafické rozhranie prešlo rôznymi iteráciami návrhmi kým sa dostalo do tejto konkrétnej podoby. Spočiatku pri návrhu som riešil len ovládanie virtuálnych objektov systémom „Drag&Drop“ s tým, že užívateľ mal možnosť len pridržať tlačidlo na obrazovke tabletu. Toto však neumožňovalo nastavenie citlivosti ovládania virtuálneho objektu a namiesto tlačidla som navrhol ovládací prvok nastavenia citlivosti umiestneného v pravej časti obrazovky tak, aby užívateľ mohol tento ovládací prvok využívať stále palcom pri držaní tabletu na šírku. Pri ovládaní objektov je potrebné, aby užívateľ mohol prepínať pohodlne rôzne metódy ovládania objektov spomedzi translácie, rotácie a zmeny merítka. Keďže užívateľ pokiaľ chce zmeniť metódu ovládania objektu musí prestať s objektom narábať (prerušit interakciu s tlačidlom citlivosti), umiestnil som tlačidlá na zmenu metód pod ovládací prvok nastavenia citlivosti a ovládania. V pravej časti obrazovky sa teda nachádzajú ovládacie prvky, ktoré sa veľmi často využívajú (napríklad pokiaľ je užívateľ pravák, pokiaľ nie, grafické rozhranie vieme jednoducho „zrkadlovo“ otočiť).

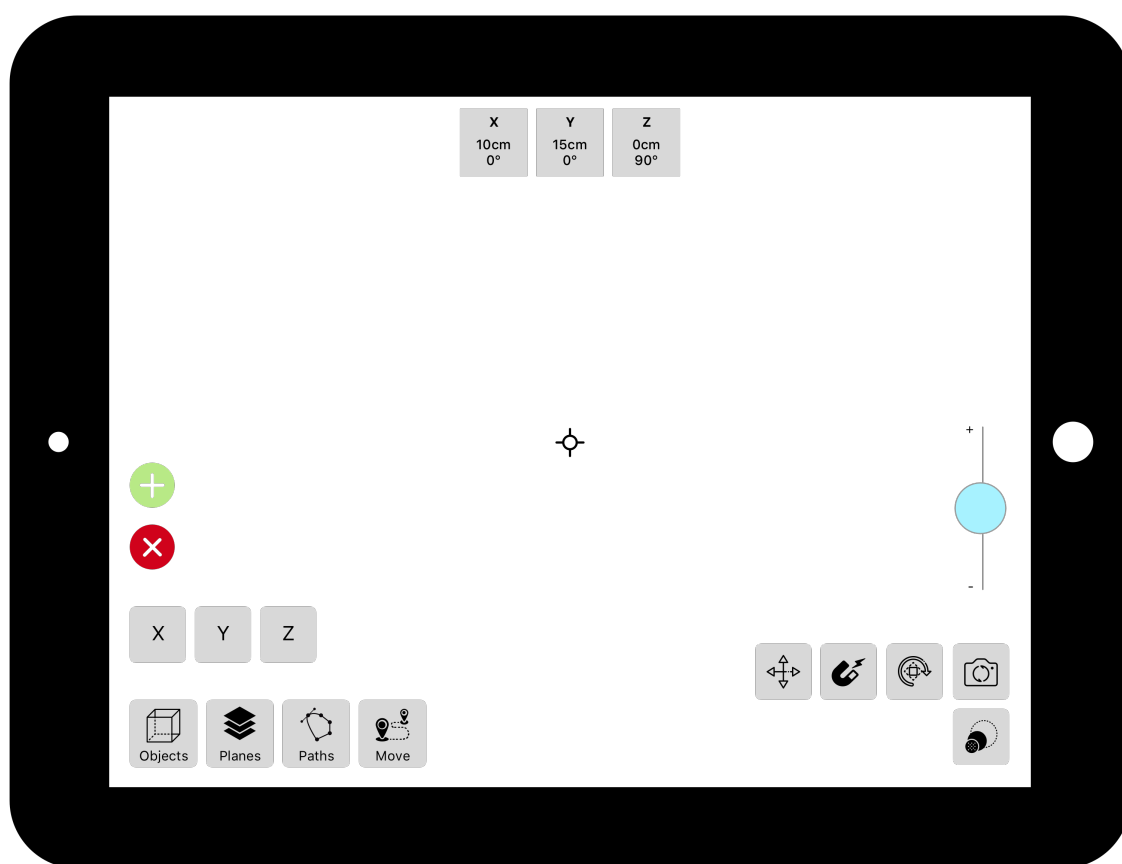
Ovládacie prvky, ktoré užívateľ môže využiť pomerne často najmä pri zostavení scény ktorú potrebuje sú umiestnené v ľavej časti obrazovky pre pohodlné ovládanie ľavým palcom. Jedná sa tak najmä o prvky vloženia a mazania objektu do scény alebo ukončenie editácie virtuálnej trajektórie.

Na základe požiadaviek na systém som navrhol tlačidlá, vďaka ktorým bude užívateľ môcť prepínať a fixovať osi X, Y, Z a povolovať/zakazovať interakciu virtuálneho objektu

cez tieto osi. Tieto tlačidlá som umiestnil na ľavú stranu obrazovky tak, aby taktiež mohol ľavým palcom pomerne často zasahovať do ovládania týchto tlačidiel a meniť tak ovládanie osí.

Poslednými prvkami sú tlačidlá, ktoré užívatelia budú využívať pravdepodobne najmenej. Sú to tlačidlá, ktoré budú mať za úlohu prepínať jednotlivé módy aplikácie. Keďže zväčša užívateľ pracuje dlhodobo v jednom móde, rozhodol som sa tieto tlačidlá umiestniť do ľavého dolného rohu obrazovky tak, aby užívateľ mal stále na tieto tlačidlá dosah avšak za „ťažších“ podmienok.

Medzi informačné prvky grafického rozhrania som navrhol využitie malého terčika umiestneného v strede obrazovky, ktorý má za úlohu informovať užívateľa o možnom mierení na objekt s ktorým môže vykonávať interaktivitu alebo možnosti umiestnenia bodu trajektórie. Keďže tento prvok by mal signalizovať možnosti či užívateľ môže vykonávať napríklad interakciu objektu alebo nie, navrhol som aby terčik menil farby medzi **červenou** a **zelenou**.



Obr. 3.3: Navrhnuté grafické rozhranie aplikácie s dôrazom na rozmiestnenie prvkov po stranách tabletu a inuitívnom využití ikon.

Ďalší informačný prvok, ktorý je veľmi významný zobrazuje informácie o polohe a rotácii objektu v osách X , Y , Z . Táto poloha je relatívna poloha voči súradnicovému systému pracovnej plochy v ktorej užívateľ pracuje. Keďže užívateľ môže pracovať samostatne v osách X , Y , Z , tento informačný prvok zobrazuje samostatne vzdialenosti a uhly v jednotlivých osách. Vzdialenosti sú uvedené v centimetroch a uhly v stupňoch a umiestnenie tohto prvku je na vrchu v strede obrazovky tak, aby užívateľovi neprekážal v pohľade na scénu a tak-

tiež aby užívateľ nemohol blokovat prstami výhľad na tento prvok. Jeho využitie je veľmi významné pri nastavovaní polohy a rotácie objektov.

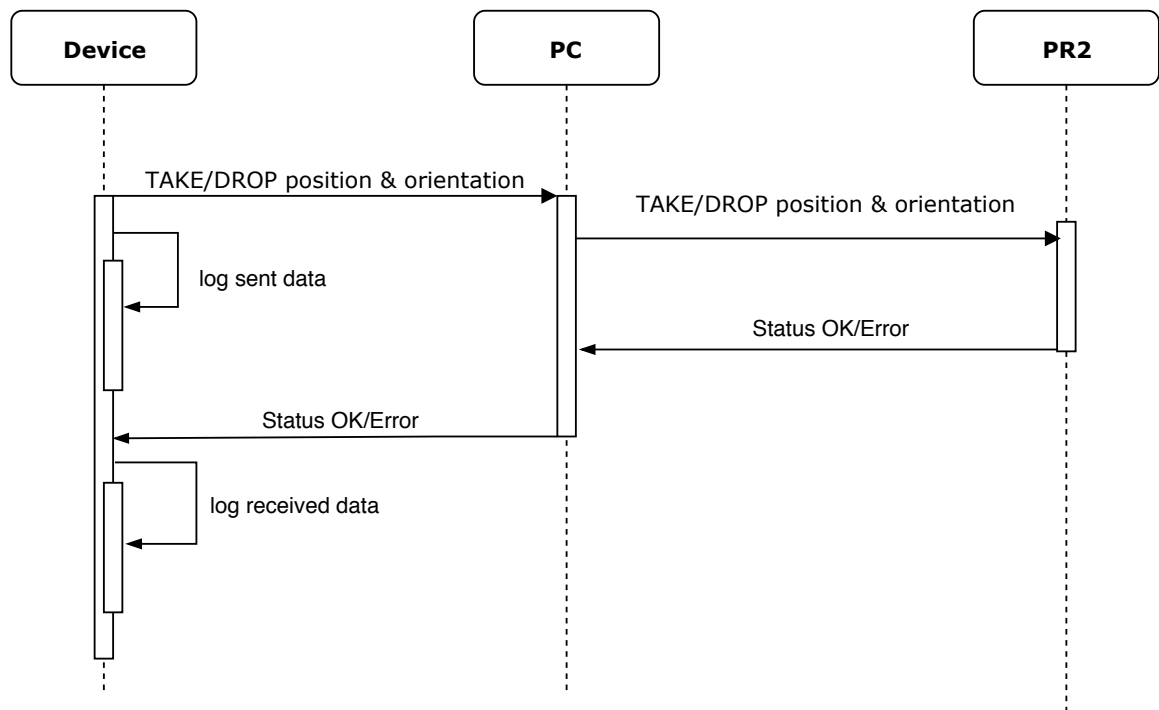
Ako vidieť na obrázku 3.3, návrh grafických prvkov rozhrania umožňuje užívateľovi prepínať módy aplikácie pre prácu s objektmi, pomocnými rovinami a tvorenie trajektórií. Pridávať objekty a mazať ich. Povoľovať a zakazovať pohyb a rotáciu cez osi X, Y, Z . A vybrať si z metód manipulácie objektov.

3.6 Komunikácia s robotickým ramenom PR2

Aby bolo možné robotické rameno PR2 ovládať, je nutné informácie z mobilného zariadenia ktoré vykonáva interakciu virtuálnych objektov nejak zdieľať s robotickým systémom a využiť to pre jeho ovládanie. Robotické rameno má množstvo vlastných súradnicových systémov v ktorých pracuje. Je nutné navrhnúť spôsob, akým zariadenie bude odosielať pozície robotickému ramenu a vyriešiť prepojenie ROS 2.5 a operačného systému iOS. V aplikácii pracujeme v oblasti pracovnej plochy, ktorá má vlastný súradnicový systém a predstavuje tak súradnicový systém robotického ramena, môžeme využiť pozíciu objektov v osiach X, Y, Z ako pozíciu, ktorú pozná aj robotické rameno a priamo ju odoslať ramenu. Existuje však viacero spôsobov, ako robotické rameno „donútiť“ k pohybu, a ako mu zadávať pozície kam sa má dostať. Vybral som jeden z dostupných spôsobov, a to využitie `ActionServer`-a na robotickom ramene, ktorý je automaticky spustený po zapnutí celého systému `ARTable`. `ActionServer` funguje na základe klient-server komunikácie, kde klient je naša aplikácia v mobilnom zariadení. Klient sa pripojí na robotické rameno pomocou identifikátora bežiackej akcie v ROS a môže mu odosielať tzv. `Goals`, teda úlohy, ktoré má robotické rameno vykonať.

Na obrázku 3.4 je znázornená komunikácia medzi zariadením, počítačom a robotickým ramenom PR2. Pre jednoduchšie posielanie pozície a orientácie objektu som navrhol využitie počítača na ktorom bude spustený `WebSocket` server, ktorý bude komunikáciu medzi zariadením a robotickým ramenom PR2 delegovať. Tento počítač slúži ako master počítač v robotickom operačnom systéme bežiacom na `ARTable` popísanom v sekcii 2.4. Skript, ktorý bude bežať na počítači bude predstavovať klienta, ktorý komunikuje s robotickým ramenom po tom, ako získa údaje o pozícii z `WebSocket` servera.

Správa, ktorú odosiela zariadenie je vo formáte `JSON` a je znázornená vo výpise 3.4. Táto správa sa odosiela vždy, keď užívateľ objekt „chytí“ s operáciou „`TAKE`“ a „pustí“ s operáciou „`DROP`“. Počítač, na ktorom bude spustený `WebSocket` server správu prijme a pokúsi sa ju rozparsovať a vytvoriť štruktúru pozície a orientácie v dátovom type pre robotické rameno. V ROS sa využíva dátový typ `PoseStamped`, ktorý má nastaviteľné parametre ako pozície tak aj orientácie.



Obr. 3.4: Diagram komunikácie aplikácie spolu s robotickým ramenom PR2, posielanie operácie, pozície a rotácie robotickému ramenu.

Po vytvorení premennej s údajmi o pozícii a rotácii ramena sa táto správa odošle ako cieľ vo vytvorenom `ActionClientovi` ktorý komunikuje s `ActionServer` na robotickom ramene. Nevýhodou tejto komunikácie je práve to, že robotické rameno často krát vyhodnotí danú pozíciu tak, že sa do nej nedostane a preto užívateľ musí využívať zobrazenie logovania odpovedí zo servera, či sa daný cieľ podarí dosiahnuť.

```

1 {
2   "position":{
3     "X":0.01,
4     "Y":0.02,
5     "Z":0.3
6   },
7   "orientation":{
8     "X":0.707,
9     "Y":0.0,
10    "Z":-0.707,
11    "W":0.0
12  },
13   "operation":"TAKE"
14 }
  
```

Kód 3.4: Návrh štruktúry správy ktorú bude posilať zariadenie počítaču pre pozíciu, orientáciu a operáciu s robotickým ramenom.

Kapitola 4

Implementácia experimentálnej aplikácie

Experimentálna aplikácia bude slúžiť na testovanie rôznych módov ovládania manipulácie objektov v rozšírenej realite. Implementoval som do aplikácie metódy ovládania objektov, ktoré som popisoval v kapitole 3 aby som tak demonštroval ich využiteľnosť ktorú následne otestujem v nasledujúcej kapitole. V tejto kapitole popíšem na akom zariadení budem aplikáciu vyvíjať, štruktúru implementácie a niektoré implementačné detaily ohľadom aplikovanie rôznych módov a metód v aplikácii.

4.1 Technický popis a realizácia

Experimentálny systém pre ovládanie virtuálnych objektov pomocou tabletu som sa rozhodol vytvoriť v operačnom systéme iOS, ktorý od verzie 11 podporuje knižnicu rozšírenej reality od Apple nazvanú **ARKit** viď sekcia 2.6 v programovacom jazyku Swift. Zariadenie, na ktorom bude aplikácia vyvíjaná je Apple iPad 5th generation s uhlopriečkou 10". ARKit umožňuje pohodlne a presne pracovať v priestore v rozšírenej realite a využíva technológiu pomenovanú ako Visual Inertial Odometry popísanú viac v sekcii 2.2. Ide o detekciu priestoru na základe detekovania významných bodov zo vstupu a kombinovaním údajov čidiel z telefónu ako je gyroskop, akcelerometer a pod.

ARKit podporuje rozoznávanie vodorovnej plochy a zo systémových požiadaviek v návrhu plyní, že je potrebné si na začiatku umiestniť vlastnú pracovnú plochu ramena. Po spustení aplikácie ARKitom sa inicializuje prostredie, kde je potrebné pohybovať so zariadením po okolí. Po detekcii prostredia je možné vyžiadať detekciu vodorovných plôch tak, aby sme získali dostatočne veľkú plochu na umiestnenie virtuálnej pracovnej plochy ramena.

Po detekcii priestoru sa umiestni počiatkový súradnicový systém na základe polohy telefónu. V tomto súradnicovom systéme môžeme pracovať s objektmi pomocou známych parametrov osí X, Y, Z . Súradnice, vzhľadom k objektu, sú v ARKite vedené v metrovej mierke. Všetky prvky ktoré vkladáme do virtuálneho priestoru využívajú tzv uzly. Uzly na seba nadväzujú a ich pozície sú relatívne k rodičovskému uzlu. Vďaka tomu je možné sem po pridaní pracovnej plochy vkladať objekty, na základe relatívnej pozície, voči uzlu danej plochy. Uzly majú teda vlastnú pozíciu X, Y, Z a aj rotáciu pomocou týchto osí.

ARKit poskytuje v reálnom čase údaje o pohybe zariadenia v priestore, jeho pozíciu a rotáciu. Vďaka týmto údajom vieme s objektmi manipulovať presne podľa pohybu a rotácie zariadenia.

Štruktúra implementácie

V aplikácii využívam dostupné API knižnice ARKit a implementujem vlastné triedy pre interakciu s objektmi, mapovanie vstupov, uloženie vytvorených objektov, prácu s virtuálnymi trajektóriami a pod. Hlavné a kľúčové časti aplikácie pre riešenie jednotlivých problémov sú:

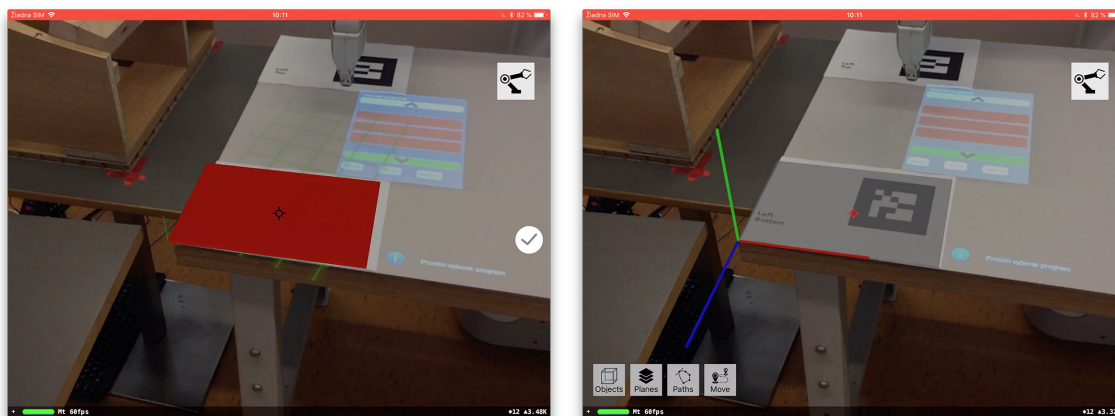
- **Spustenie aplikácie a umiestnenie pracovnej plochy** – Rieši nastavenie `ARSession` pomocou konfigurácie `ARWorldTrackingConfiguration` v triede `ViewController`.
- **Vytváranie všetkých virtuálnych objektov** – Vytvorená trieda `WSVirtualObjects`, ktorá je vytvorená ako *singleton* a poskytuje vytvorenie objektu, mazanie objektu, vytvorenie virtuálnych trajektórií a ich editáciu, uloženie pomocných rovín a iné.
- **Manipulácia s virtuálnymi objektmi** – Trieda nazvaná `VirtualObjectInteraction`, ktorá umožňuje veškerú interakciu s virtuálnymi objektmi, ich transláciu, rotáciu, zmenu merítka, sú v nej implementované metódy ovládania objektov, označovanie objektov a vyhľadávanie objektov na základe `hitTest` metódy.
- **Grafické prvky rozhrania** – Grafické prvky rozhrania sú rozdelené do jednotlivých častí pre ovládanie módov, metód, osí, citlivosti a podobne. Všetky tieto rozhrania sú navrhnuté grafickým editore s príponou `.xib`. Sú to napríklad rozhrania `InteractionMovementModeView.xib` pre ovládanie metód interakcie, `InteractionAxisOptionsView.xib` pre ovládanie osí *X, Y, Z*, a ďalšie.
- **Komunikácia s WebSocket** – V triede `SocketClientSingleton` implementujem jednoduchý `WebSocket` klient, ktorý sa pripojí na danú IP adresu a port a následne odosiela prijmané pozície objektu z triedy `VirtualObjectInteraction` s využitím buffera na odosielanie.

4.2 Súradnicový systém a pracovná plocha

Každé prvé spustenie aplikácie či už z pozadia alebo nie, musí vykonať detekciu a vyhľadávanie vodorovnej dostatočne veľkej plochy pre našu prácu. Detekovanie pracovnej plochy vykonáva ARKit sám na základe konfigurácie uloženej

v `ARWorldTrackingSessionConfiguration` pomocou premennej `planeDetection`. Nastavenie detekcie horizontálnej plochy a inicializáciu ARKitu pomocou `ARSession` prebieha pri spustení aplikácie po nastavení veľkosti pracovnej plochy.

Po umiestnení virtuálnej pracovnej plochy sa pomocou transformačnej matice nastaví globálny súradnicový systém do ľavého dolného rohu plochy. Nakoľko ARKit môže detekovať plochu o rôznej veľkosti s rôznym natočením, spravil som možnosť rotácie pracovnej plochy gestami pre správne umiestnenie plochy podľa potreby so znázorneným súradnicovým systémom.



Obr. 4.1: Na obrázku vľavo je znázornené umiestňovanie pracovnej plochy (aktívne pulzovanie červenej farby) do detekovaného priestoru (zelená mriežka). Na obrázku vpravo je umiestnená pracovná plocha o rozmere $20 \times 30\text{cm}$ spolu s nastavením súradnicového systému.

```

1 // Get transformation matrix & change X,Z position to left-
  bottom corner of Working Space
2 var worldTransform = workingSpace.simdWorldTransform
3 worldTransform.columns.3.x -= Float(DefaultValues.shared.
  workingSpaceSize.width / 200)
4 worldTransform.columns.3.z += Float(DefaultValues.shared.
  workingSpaceSize.height / 200)
5
6 self.sceneView.session.setWorldOrigin(relativeTransform:
  worldTransform)

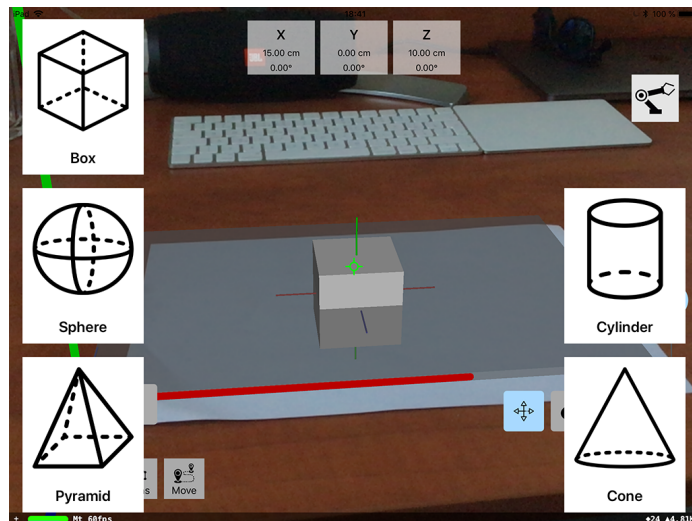
```

Kód 4.1: Nastavenie súradnicového systému podľa umiestnenej pracovnej plochy v priestore a využitia transformačnej matice.

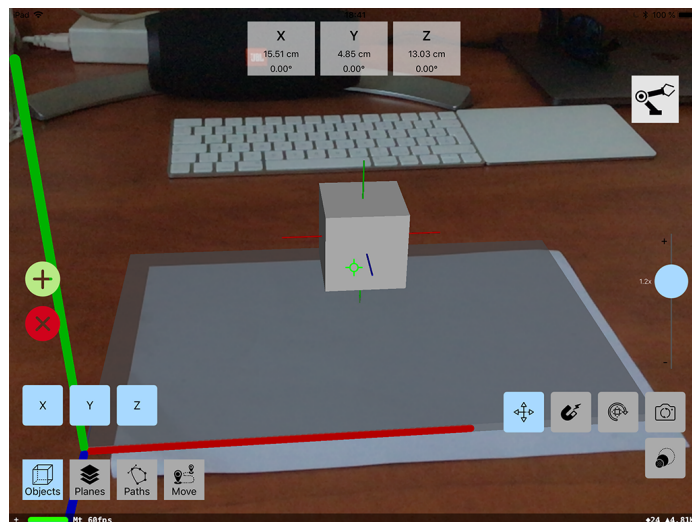
4.3 Implementácia módov systému

Ako som popisoval v návrhu aplikácie, systém sa skladá zo štyroch hlavných módov aplikácie. Je to mód pre ovládanie virtuálnych objektov, mód na prácu s pomocnými rovinami a dva módy na tvorbu trajektórie robotického ramena. Každý tento mód v aktívnom stave má mierne pozmenené grafické rozhranie oproti návrhu, kde skryjem nepotrebné prvky a zobrazím tie, ktoré bude mód využívať. Nižšie zobrazujem implementovaný stav užívateľského rozhrania pri používaní daných módov.

Na obrázku 4.2 a 4.3 je znázornené implementované užívateľské rozhranie spolu s dostupnými objektmi ktoré môže užívateľ vytvoriť, dostupnými dvoma metódami pre transláciu objektu, dvoma metódami rotácie objektu a metódu pre zmenu merítka objektu. Podrobnejší popis implementácie týchto metód je popísaný v sekcii 4.4. Mód ovládania objektov má nad tlačidlami metódami ovládania objektov polohovacie tlačidlo na zmenu citlivosti pri manipulácii s objektom. Na týchto obrázkoch vidieť aj zelenú signalizáciu nájdenia objektu v terčíku, a zobrazenie osí X, Y, Z označeného objektu.



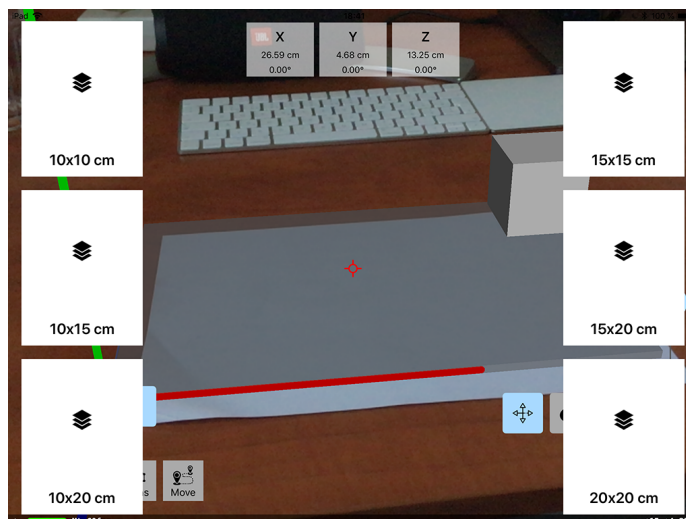
Obr. 4.2: Grafické rozhranie aplikácie pri pridávanie virtuálneho objektu v móde manipulácie objektov.



Obr. 4.3: Mód manipulácie objektov, ktorý obsahuje 2 rôzne spôsoby translácie objektu, 2 spôsoby rotácie obejktu a mód pre zmenu merítka objektu.

Hľadanie objektov sa realizuje pomocou metódy `hitTest(_:types:)`. Metóda sa volá zakaždým keď renderer deleguje `ARFrame` popísaný v sekcii ?? do triedy `VirtualObjectsInteraction`, čo umožňuje neustále prehľadávať priestor na ktorý sa užívateľ pozerá a hľadať v ňom objekty záujmu.

Pri móde pridávania a manipulácie s pomocnými rovinami sa v grafickom rozhraní prvky nemenia – s pomocnými rovinami pracujeme tak isto ako s virtuálnymi objektmi s tým rozdielom, že môžeme namiesto pridania virtuálnych objektov do priestoru pridať pomocné roviny viď obrázok 4.4.



Obr. 4.4: Grafické rozhranie aplikácie pri pomocných rovin do priestoru s preddefinovanými veľkosťami.

Tvorba virtuálnej trajektórie pomocou bodov spojených priamkou je vytvorená ako zoznam s počiatočným a koncovým bodom v priestore a ich spojnicou, znázornené vo výpise 4.2. CustomSCNNode je vytvorený vlastná trieda pre vytvorenie uzla v ARKite, ktorá dedí vlastnosti z triedy SCNNode.

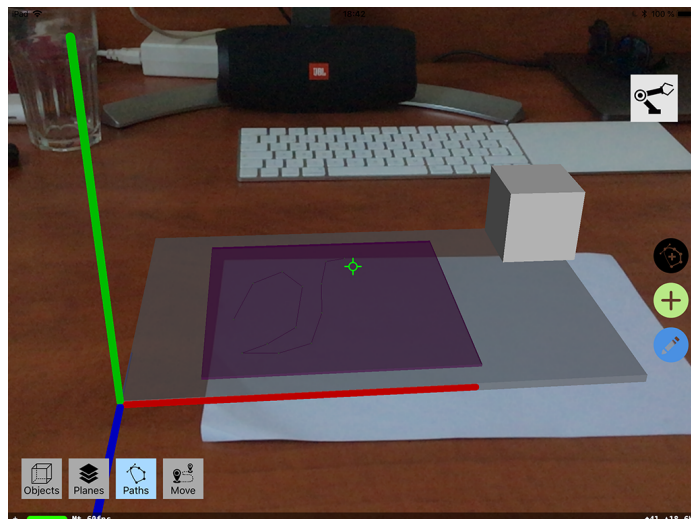
```

1 public struct PathPoint {
2     var startPoint : CustomSCNNode!
3     var endPoint : CustomSCNNode?
4     var line : CustomSCNNode?
5 }

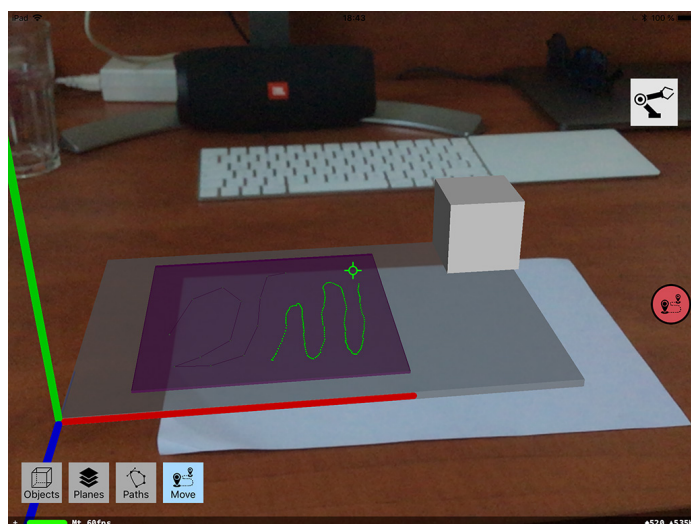
```

Kód 4.2: Štruktúra jedného bodu zoznamu pre tvorenie vlastnej trajektórie pomocou priamiek.

Pri editácii virtuálnej trajektórie môže užívateľ s bodmi manipulovať tak, ako pri manipulácii virtuálnych objektov. Na obrázkoch 4.5 a 4.6 je znázornené grafické rozhranie aplikácie pri módoch trajektórie voľným pohybom a trajektórii pomocou priamiek.



Obr. 4.5: Grafické rozhranie aplikácie pri tvorbe trajektórie pomocou bodov spojených priamkami. Čierne tlačidlo vytvára novú trajektóriu do ktorej následne zeleným tlačidlom plus môže užívateľ pridávať body. Modrým tlačidlom užívateľ aktivuje mód editácie trajektórie.



Obr. 4.6: Grafické rozhranie aplikácie pri tvorbe trajektórie voľným pohybom. Postačuje jedno tlačidlo, ktoré spustí a vypne nahrávanie voľného pohybu počas ktorého sa vkladajú body na pomocnú rovinu.

4.4 Implementácia metód pre manipuláciu virtuálnych objektov

Keďže ARKit (2.6) poskytuje údaje polohy zariadenia v súradnicovom systéme, orientácie a rotácie zariadenia v reálnom čase pomocou **ARFrame**, implementované metódy využívajú všetky tieto údaje k svojej funkčnosti. Pri začatí manipulácie s objektmi sa uloží aktuálna poloha zariadenia do štruktúr **sPosition** a **sRotation** znázornených vo výpise 3.1.

Implementácia metódy priamej manipulácie

Priama manipulácia objektu mapuje pohyb zariadenia v priestore s využitím citlivosti (hodnoty premennej `mSensitivity`), aplikovanie tejto metódy pre os X (`mAxisOpt.x`) s aktualizovaním počiatočnej pozície zariadenia (`sPosition.x`) je znázornené na výpise 4.3.

```
1 if mAxisOpt.x == true && sPosition.x != nil {  
2     let value = (sPosition.x - cameraPosition.x) * mSensitivity  
3     selectedObject.position.x -= value  
4     sPosition.x = cameraPosition.x  
5 }
```

Kód 4.3: Časť kódu pre priamu manipuláciu objektu pre os X s využitím citlivosti ovládania.

Implementácia metódy tiaha

Metóda tiaha je jednoduchá metóda ovládania pohybu, ktorá upravuje pozíciu objektu v kladnom alebo zápornom smere cez jednu aktívnu os z osí X, Y, Z . Aplikovanú metódu vidieť vo výpise 4.4.

```
1 // adjust for centimetres and positive/negative value  
2 let movingIn = (mSensitivity - 1) / 100  
3  
4 if mAxisOpt.x == true {  
5     selectedObject.position.x = sPosition.x + movingIn  
6 }
```

Kód 4.4: Časť kódu pre priamu manipuláciu objektu pre os X .

Implementácia metódy rotácie pomocou zariadenia

Spolu s aktualizovaním priestoru a využitím rotácie zariadenia z objektu `ARFrame` vieme určiť o akú hodnotu zariadenie rotovalo v určitom smere a následne túto rotáciu aplikovať priamo na objekt. Jedná sa o veľmi podobnú metódu mapovania dostupných vstupov na objekt (znázornenú vo výpise 4.5 ako pri priamej manipulácii s objektom). Objekt rotuje pomocou rozdielu uhlov v počiatočnom stave zariadenia `sRotation` a aktuálnom stave zariadenia `cameraRotation` s využitím citlivosti `mSensitivity`.

```
1 if mAxisOpt.x == true && sRotation.xAngle != nil {  
2     selectedObject.eulerAngles.x -= (sRotation.xAngle -  
3         cameraRotation.x) * mSensitivity  
4     sRotation.xAngle = cameraRotation!.x  
5 }
```

Kód 4.5: Časť kódu ktorá ukazuje mapovanie zmeny rotácie zariadenia na objekt a tým vytvára rotáciu objektu cez os X .

Implementácia metódy rotovania pomocou kosínusovej vety

Algoritmus metódy vypočíta veľkosti úsečiek vo vytvorenom trojuholníku pozostávajúceho z polohy objektu, štartovacej a aktuálnej pozície zariadenia pomocou pytagorovej vety.

Po vypočítaní úsečiek vypočíta uhol funkcia `calculateAngle(a, b, c)` znázornená vo výpise kódu 4.6, ktorá vracia veľkosť vypočítaného uhla pomocou kosínusovej vety. Tento uhol sa mapuje priamo na virtuálny objekt.

```
1 // Calculate angle by cosinus sentence
2 func calculateAngle(size_A a : Double, size_B b : Double,
   size_C c : Double) -> Float {
3   let top = a*a + b*b - c*c
4   let bottom = 2 * a * b
5   let result = top / bottom
6   return Float(acos(result))
7 }
```

Kód 4.6: Implementácia funkcie pre výpočet uhla pomocou kosínusovej vety.

Kapitola 5

Testovanie experimentálnej aplikácie

Navrhnuté postupy je nutné otestovať a vyhodnotiť na úlohách s reálnymi užívateľmi. Popíšem faktory testovania, ktoré kľúčové veci bolo dôležité testovaním sledovať, testovaciu scénu ktorú užívatelia mali vyskladať a na záver zhodnotím testovanie spolu s pozorovaním užívateľov ktorý aplikáciu skúšali na konferencii Excel@FIT.

5.1 Popis a faktory testovania na užívateľoch

V prvom rade som chcel zistiť mierne nedostatky a nefunkčnosti aplikácie. Nebolo to však až tak dôležité zisťovať, no poslúžilo to na testovanie metód manipulácie z hľadiska funkčnosti. Testovaním na používateľoch som zisťoval to, ako užívatelia s aplikáciou pracujú, od jej zapnutia, detekciu scény a následnú manipuláciu s objektmi, pomocnými rovinami a trajektóriami. Počas testovania metód sa nám naskytne možnosť nových pohľadov na prácu s virtuálnymi objektmi, alebo možnosť zistiť ktoré činnosti sú v aplikácii nejasné, prípadne ktoré ovládacie prvky sú zle umiestnené.

Faktory testovania

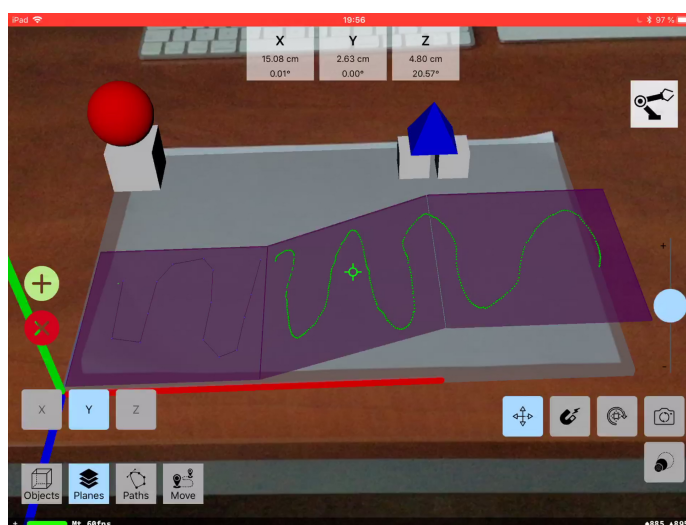
Dôležité faktory, ktoré chcem na užívateľoch sledovať sú tie, ako s danými objektmi manipulujú. Porovnať si metódy zaobchádzania s objektmi, translácie a rotácie z hľadiska faktorov ako sú presnosť, rýchlosť, zrozumiteľnosť. Potvrdiť alebo vyvrátiť navrhovanie metód z kapitoly návrhu. A v neposlednom rade samozrejme ako aplikácia pôsobí na užívateľa, či rozumie UI prvkom a ich rozložením, správne držaniu tabletu alebo význam pomocných rovin a trajektórií.

Navrhujem merať to, ako užívateľ rozumie grafickému rozhraniu čistým pozorovaním užívateľa ako s aplikáciou pracuje, ako zariadenie drží a či grafické ovládacie prvky vie pohodlne ovládať. Pre vyhodnotenie grafického rozhrania sa budem užívateľov pýtať, či grafickému rozhraniu rozumeli a ktoré prvky grafického rozhrania sú pre nich nezrozumiteľné. Ďalej by som chcel otestovať metódy manipulácie objektov približným meraním dĺžky nastavenia translácie a rotácie objektu a pozorovať, ako sa s danou metódou užívateľovi pracuje.

Priebeh testovania a testovacia scéna

Testovanie rozdelím do dvoch častí, kde v prvej časti testovania poskytnem užívateľovi tablet so spustenou aplikáciou a následne budem sledovať to, ako bude ovládacie prvky využívať a ako s nimi bude narábať. Touto časťou testovania sa zamerám na zrozumiteľnosť užívateľského rozhrania a aplikácie ako celku.

Druhou časťou testovania je zostavenie testovacej scény podľa predlohy ktorú užívateľ uvidí. Užívateľovi poviem, ktoré ovládacie prvky na čo slúžia. Testovaciu scénu som zostavil z niekoľkých virtuálnych objektov, troch pomocných rovín umiestnených tak, aby boli jednou hranou spojené a pomocná rovina v strede bola naklonená o určitý uhol v jednej osi a virtuálnych trajektórií umiestnených na tieto pomocné roviny, viď obrázok 5.1. Touto časťou testovania sa budem snažiť testovať navrhnuté metódy ovládania objektov, ktoré užívateľ bude musieť využiť pri vytváraní scény.



Obr. 5.1: Testovacia scéna, ktorú užívateľ musel vyskladať. Scéna využíva možnosti narábania s objektmi, pomocnými rovinami a tvorenie trajektórie. Testovaním sledujem efektívnosť navrhnutých metód ovládania virtuálnych objektov.

5.2 Testovanie

Testovanie prvej časti bolo vykonané počas študentskej konferencie Excel@FIT¹ na približne 30tich užívateľoch, ktorí mali záujem si aplikáciu vyskúšať. V druhej časti testovania mali užívatelia za účel vyskladať scénu podľa predlohy a testovať tak navrhnuté metódy.

Porozumenie GUI

Pri práci s aplikáciou som pozoroval, ako užívateľ s grafickým rozhraním aplikácie pracuje. Toto testovanie som pozoroval pri oboch skupinách užívateľov, a dal väčšiu prioritu na odozvu od užívateľov, ktorým som grafické rozhranie aplikácie nevysvetľoval podrobne. Prekvapilo ma, že užívatelia ktorým som grafické rozhranie aplikácie popísal s ním vedeli okamžite pracovať bez väčších problémov. Užívatelia, ktorým som aplikáciu nepredstavoval

¹<http://excel.fit.vutbr.cz>

mali vždy tendenciu preklikať tlačidlá módov a metód so systémom „Čo to spraví, keď to stlačím?“. Niektorí užívatelia však mali vždy potrebu využívať dotykový displej tabletu a stláčať rôzne po obrazovke tam, kde bola umiestnená pracovná plocha aby napríklad umiestnili na plochu objekt.

Manipulácia objektov

Užívateľom v prvej časti testovania som metódy manipulácie nijak nevysvetloval a pozoroval, ako budú s objektmi manipulovať. Keďže žiadny z užívateľov nevedel, aké metódy manipulácie sú dostupné, riadil sa intuitívnym postupom na základe vlastného uváženia. Väčšina ľudí sa vždy snažila pohnúť s objektom vykonaním rotácie zariadenia a mala tak predstavu pripnutého objektu k zariadeniu a po rotácii zariadenia očakávali, že objekt sa premiestni tam, kde ukazuje terčik umiestnený v strede displeja. Po odskúšaní dostupných metód manipulácie užívatelia testovali ich využiteľnosť.

Metóda priamej manipulácie bola pre ľudí zväčša intuitívna po vysvetlení, ako funguje, že objekt odráža presne pohyb zariadenia. Pri tejto metóde užívatelia nechápali celkom význam využitia fixácií osí X, Y, Z , keďže po zvolení metódy sú všetky 3 osi deaktivované a objekt sa pri manipulácii nehýbe. Po aktivovaní osí X, Y, Z užívatelia skúšali hýbať s objektom na základe zmeny polohy zariadenia v priestore, avšak vôbec nevyužívali prvok citlivosti. Až po vysvetlení, ako prvok na ovládanie citlivosti funguje sa ho snažili niekoľko krát použiť vrámci testovania pohybu objektu. Užívatelia využívali túto metódu skôr k veľkým pohybom objektu o niekoľko desiatok centimetrov a keď mali za úlohu jemnejšie narábať s objektom, využili radšej metódu tiahla popísanú ďalej.

Metóda tiahla je metóda, ktorá vykonáva transláciu objektu len do niekoľko centimetrov v kladnom/zápornom smere s presnosťou na milimetre a využíva na tento pohyb len zmenu polohy prvku na ovládanie citlivosti. Túto metódu si užívatelia volili na presnejšie ovládanie objektu a o mierne posuny podľa jednej z aktivovaných osí.

Rotovanie objektu pomocou kosínusovej vety bola metóda, ktorá sa mi zdala veľmi intuitívna, avšak pre užívateľov to pôsobilo práve naopak. Užívatelia nevedeli, podľa ktorej z osí bude objekt rotovať a chýbalo im nejaké vyznačenie tejto rotácie. Čo však užívatelia ocenili bolo to, že po aktivovaní všetkých 3 osí sa objekt natáčal vždy smerom k polohe zariadenia a boli schopní tak objekt rotovať aj napríklad o viac ako 40° . Čo sa týka presnosti nastavenia uhlov objektu, užívatelia vedeli narábať s ovládacím prvkom citlivosti a „zjemňovať“ tak rotáciu objektu na požadované hodnoty.

Rotovanie objektu pomocou rotácie zariadenia bola pre užívateľov veľmi intuitívna metóda, keďže hneď zo začiatku mali tendenciu s objektom rotovať na základe rotácie zariadenia ako napríklad v hrách alebo iných aplikáciách. Pre presnejšie nastavenie požadovaného uhla využívali užívatelia taktiež nastavenie citlivosti a metóda im nespôbovala žiadne problémy v ovládaní. Jedinou nevýhodou, ktorú užívatelia považovali za dosť podstatnú bolo to, že pri rotovaní naraz o väčší uhol mohol objekt zmiznúť z obrazovky zariadenia a užívateľ tak stratil prehľad o tom, či objekt rotuje správne. Toto sa však podarilo do istej miery vyriešiť tým, že užívatelia využili nastavenie citlivosti rotácie alebo rotáciu vykonávali na viac krát.

Zmena merítka objektu si nevyžadovala veľkú pozornosť nakoľko táto metóda bola pre všetkých užívateľov úplne intuitívna a pochopiteľná na prvý krát.

Pomocné roviny a trajektórie

Užívatelia, ktorí nevedeli na čo pomocné roviny v aplikácii slúžia bádali, ako body trajektórie ukladať napríklad na pracovnú plochu. Tento mnou navrhnutý spôsob využitia pomocných rovín bolo potrebné užívateľom vysvetliť. Po vysvetlení ako pomocné roviny pridať a na čo slúžia vedeli užívatelia plnohodnotne s nimi pracovať ako pri manipulácii objektov a využívali všetky metódy manipulácie vďaka ktorým pomocné roviny nastavili podľa predlohy scény ktorú mali zostaviť. Zaujímavé však bolo sledovať ako si poradia s rotáciou pomocnej roviny. Takmer všetci užívatelia si vybrali metódu rotovania zariadenia a nie metódu s využitím kosínusovej vety, nakoľko sa im táto metóda zdala intuitívnejšia a jednoduchšia. Užívatelia následne pridávali trajektóriu pomocou bodov umiestnených na pracovné roviny a nahrávaním voľného pohybu s čím nemali väčší problém. Veľmi ocenili využitie terčíka pri ukladaní bodov a taktiež jednoduchosť pridávania nových trajektórií na pomocné roviny.

Zhodnotenie metód

Na základe pozorovania a slovnej odozvy od užívateľov som vytvoril tabuľku 5.1 popisujúcu využitie jednotlivých metód na rôzne spôsoby ovládania. V tabuľke zhodnocujem metódy podľa toho, ako užívatelia ohodnotili ich využiteľnosť z hľadiska veľkých prírastkov translácie alebo rotácie, presnejšieho pohybu s vysokou citlivosťou a aj podľa toho, či metóda pôsobila intuitívne.

	Priama manipulácia	Tiahlo	Kosínusová veta	Rotovanie zariadenia
veľké prírastky	Áno	Nie	Áno	Nie
presnejšie ovládanie	Nie	Áno	Nie	Áno
intuitívna metóda	Áno	Áno	Nie	Áno

Tabuľka 5.1: Tabuľka porovnáva navrhnuté metódy translácie a rotácie objektov na základe užívateľského testovania.

Navrhnuté metódy ktoré užívatelia testovali by pôsobili úplne inak, pokiaľ by som v aplikácii nenavrhol prvok na ovládanie citlivosti manipulácie. Tento ovládací prvok umožňuje takmer vo všetkých prípadoch rýchlejšie narábanie s objektom a stáva sa dôležitou súčasťou užívateľského rozhrania aplikácie.

Metóda priamej manipulácie nebola vyhodnotená ako metóda, ktorá by slúžila na presnejšie ovládanie objektu, aj keď sa dá s touto metódou pomerne presne pracovať. Ďalšia metóda na transláciu objektu umožňovala **tiahlo** meniť polohu objektu v jednom smere po aktívnej osi a tak nie je vhodná na veľké prírastky pohybu, táto metóda je však **presnejšia** pri nastavovaní rôznych detailov polohy objektu. Metóda rotácie pomocou **kosínusovej vety** umožňuje veľké prírastky rotácie na základe zmeny polohy zariadenia pri narábaní s objektom, no užívatelia ju vyhodnotili ako skôr nepoužiteľnú než použiteľnú. Najlepšie na tom obstála metóda **rotácie objektu pomocou rotácie zariadenia**, kde užívatelia

vedeli túto metódu intuitívne automaticky použiť nakoľko takéto ovládanie je dosť používané v hrách na mobilných zariadeniach a iných aplikáciách. **Nevýhodou** tejto metódy je, že pokiaľ chce užívateľ rotovať objekt o väčší uhol, je dosť pravdepodobné, že mu tento objekt zmizne z dohľadu a nebude tak môcť kontrolovať celý kontext rotovania. Taktiež nie je moc pohodlné rotovať objekt o väčšie uhly rotovaním zariadenia nakoľko užívateľ stráca úplne prehľad o scéne.

5.3 Zhodnotenie testovania

Testovanie aplikácie by som hodnotil veľmi pozitívne nakoľko som získal lepší pohľad na to, ako manipuláciu s objektmi vidia užívatelia. Z metód ktoré som navrhol a v aplikácii implementoval by som považoval všetky za využiteľné avšak niektoré menej, niektoré viac. Pri priamej manipulácii objektov užívateľom vyhovovalo to, že objekt odráža pohyb zariadenia a vedeli s touto metódou okamžite pracovať aj s využitím citlivosti. Metóda tiahla si nenašla až také významné využitie nakoľko užívatelia vedeli doladiť miernu presnosť aj pomocou metódy priamej manipulácie. Rotovanie objektov pomocou zmeny zariadenia v priestore je metóda, ktorú užívatelia neradi využívajú a namiesto toho rotovali objekt priamo rotáciou zariadenia. Priama rotácia zariadenia nesie hlavnú nevýhodu v tom, že sa objekt stratí z dohľadu pri väčšom nastavovaní uhla a užívateľ tak stráca prehľad o tom, či objekt rotuje správne. Toto sa dá však vykompenzovať tým, že užívateľ bude prerušovať rotáciu alebo využívať ovládací prvok na citlivosť.

Čo sa týka pomocných rovín, užívatelia s nimi nemali problém manipulovať keďže sa jednalo o tie isté spôsoby manipulácie ako pri virtuálnych objektoch. Následne užívatelia mohli pridávať na pomocné roviny body trajektórie veľkým pohybom alebo bodov spojených priamkami. Oba tieto spôsoby tvorenia trajektórie boli pre užívateľov intuitívne a terčík umiestnený v strede obrazovky im dal spätnú väzbu kde môžu body trajektórie pohybu ukladať.

Aj napriek tomu, že testovanie neprebehlo na veľkej vzorke užívateľov a všetci užívatelia boli technicky zdatní (jednalo sa o študentov alebo pracovníkov vysokej školy Vysoké učení technické v Brně), získal som množstvo pozitívnych informácií ktoré by sa dali využiť v budúcnosti pri navrhovaní ďalších metód manipulácie.

Kapitola 6

Záver

Cieľom diplomovej práce bolo navrhnutie a vytvorenie spôsobov ovládania virtuálnych objektov v rozšírenej realite. Práca predstavuje rozšírenú realitu z pohľadu použiteľnosti v existujúcich riešeniach a nové technológie vďaka ktorým môžeme programovať aplikácie podporujúce rozšírenú realitu na mobilných zariadeniach. Bolo potrebné navrhnuť metódy manipulácie virtuálnych objektov pri ovládaní robotického ramena pomocou rozšírenej reality a tabletu. Vyriešiť problémy, ktoré sa manipulácie objektov týkajú a navrhnuť komplexný systém, ktorý by splňoval požiadavky.

Navrhol som experimentálnu aplikáciu s dvoma metódami translácie objektov a dvoma pre rotáciu objektov. Tieto metódy som implementoval v experimentálnej aplikácii a následne testoval s užívateľmi. Aplikácie mimo manipulácie objektov slúži aj k vytvoreniu trajektórie robotického ramena pomocou navrhnutých pomocných rovín a prináša tak viac módov spôsobu interakcie ramena s tabletom. Navrhol som spôsob komunikácie s robotickým ramenom PR2, ktoré vykonávalo pohyb manipulácie objektov na základe spoločného súradnicového systému robotického ramena a scény v rozšírenej realite. Demonštrácia ovládania robotického ramena tvorí základ pre možné rozšírenia a využitia aplikácie v priemysle v malých podnikoch.

Riešenie bolo testované niekoľkými užívateľmi napríklad na študentskej konferencii Excel@FIT a ukázalo sa, že niektoré navrhnuté metódy nepôsobia spočiatku intuitívne alebo nie sú veľmi využiteľné pri manipulácii objektov. Užívateľia priniesli aj nové, zaujímavé pohľady na riešenie aplikácie ktoré by sa dali do budúcnosti využiť.

Prácu považujem za veľmi prínosnú z hľadiska návrhu možných využití v budúcnosti. Rád by som výsledky tejto práce využil v spolupráci s výskumnou skupinou Robo@FIT na demonštrácii využitia robotického ramena v priemysle a rozvíjal túto problematiku naďalej. Existuje množstvo rôznych možností, ako s aplikáciou pokračovať. Jedna z možností je, aby robotické rameno a systém ROS mal údaje o objektoch ktoré sa nachádzajú v pracovnej ploche ramena a tieto objekty by boli zdieľané aj mojou aplikáciou, išlo by teda o kombináciu virtuálnych a reálnych objektov.

Literatúra

- [1] Apple: *Introducing ARKit: Augmented Reality for iOS*. [Online; navštívené 7.5.2018; Session 602].
URL <https://developer.apple.com/videos/play/wwdc2017/602/>
- [2] Atheer: *How Companies Are Using Augmented Reality Software and Hardware Today*. [Online; navštívené 7.01.2018].
URL <http://atheerair.com/value-of-augmented-reality/>
- [3] Azuma, R. T.: *A survey of augmented reality*. MIT Press, August 1997, presence: Teleoperators and Virtual Enviroments 6, 4.
- [4] Gamet, J.: *Favorite iOS 11 ARKit Apps for 2017*. [Online; navštívené 27.12.2017].
URL <https://www.macobserver.com/news/favorite-ios-11-arkit-apps-2017/>
- [5] Hachet, M.; Guitton, P.; Reuter, P.: The CAT for Efficient 2D and 3D Interaction As an Alternative to Mouse Adaptations. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST '03*, New York, NY, USA: ACM, 2003, ISBN 1-58113-569-6, s. 225–112, doi:10.1145/1008653.1008689.
URL <http://doi.acm.org/10.1145/1008653.1008689>
- [6] Harris, C.; Stephens, M.: *A combined corner and edge detector*. *Proceedings of the Alvey Vision Conference*, 1988: s. 147–152.
- [7] Henrysson, A.; Marshall, J.; Billinghurst, M.: Experiments in 3D Interaction for Mobile Phone AR. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia, GRAPHITE '07*, New York, NY, USA: ACM, 2007, ISBN 978-1-59593-912-8, s. 187–194, doi:10.1145/1321261.1321295.
URL <http://doi.acm.org/10.1145/1321261.1321295>
- [8] Henrysson, A.; Ollila, M.; Billinghurst, M.: Mobile Phone Based AR Scene Assembly. In *Proceedings of the 4th International Conference on Mobile and Ubiquitous Multimedia, MUM '05*, New York, NY, USA: ACM, 2005, ISBN 0-473-10658-2, s. 95–102, doi:10.1145/1149488.1149504.
URL <http://doi.acm.org/10.1145/1149488.1149504>
- [9] Hyypä, E.: *Where fashion meets technology*. [Online; navštívené 8.1.2018].
URL <http://www.paperwhite-studio.com/work/pictofit/>
- [10] Li, P.; Qin, T.; Hu, B.; aj.: *Monocular Visual-Inertial State Estimation for Mobile Augmented Reality*. *Robotics Institute, Hong Kong University of Science and*

- Technology.
 URL <http://www.ece.ust.hk/~eeshaojie/ismar2017peiliang.pdf>
- [11] Lukas, B.; Kanade, T.: *An iterative image registration technique with an application to stereo vision. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1981: s. 674–679.
 - [12] Magellium: *Augmented Reality (Energy and Defence industrials – since 2011)*. [Online; navštívené 7.01.2018].
 URL <http://www.magellium.com/blog/portfolio-item/augmented-reality/>
 - [13] Materna, Z.; Kapinus, M.; Beran, V.; aj.: Using Persona, Scenario, and Use Case to Develop a Human-Robot Augmented Reality Collaborative Workspace. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, HRI '17*, New York, NY, USA: ACM, 2017, ISBN 978-1-4503-4885-0, s. 201–202, doi:10.1145/3029798.3038366.
 URL <http://doi.acm.org/10.1145/3029798.3038366>
 - [14] Méri, A.: *Introducing New GPS Navigation Feature: Real View Navigation*. [Online; 3.07.2017].
 URL <http://www.sygic.com/blog/2017/introducing-new-gps-navigation-feature-real-view-navigation>
 - [15] Nistér, D.: *An efficient solution to the five-point relative pose problem. IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 6, 756–777, 2004.
 - [16] OpenCV: *FAST Algorithm for Corner Detection*. [Online; navštívené 10.1.2018].
 URL https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_fast/py_fast.html
 - [17] Rosten, E.; Drummond, T.: *Machine learning for high-speed corner detection. European Conference on Computer Vision*, 2006: s. 430–443.
 - [18] Schmalstieg, D.; H, T.: *Augmented reality: principles and practice*. Addison-Wesley, 2016, ISBN 0-321-88357-8.
 - [19] Sixty.no: *Augmented reality set for TV 2's World Women's Handball Championship broadcast*. [Online; 9.12.2015].
 URL http://www.vizrt.com/casestudies/47173/Augmented_reality_set_for_TV_2s_Womens_Handball_Championship_broadcast
 - [20] Tomasi, C.; Kanade, T.: *Detection and tracking of point features. Shape and motion from image streams: A factorization method–Part 3. Technical Report CMU-CS-91-132, School of Computer Science, Carnegie Mellon University*, 1991.
 - [21] Trends, A. R.: *Role of Augmented Reality in Medical Industry*. [Online; navštívené 7.01.2018].
 URL <http://www.augmentedrealitytrends.com/augmented-reality/medical-augmented-reality.html>
 - [22] Vk, V.: *Best iPhone Translation Apps for Travellers That Work Offline & Without Internet*. [Online; navštívené 7.01.2018].
 URL <https://www.iosapps4u.com/best-iphone-translation-apps/>